Optimization Based Control for Multi-agent System with Interaction

by

Cheng Peng

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor Roberto Horowitz
Assistant Professor Anil Aswani

Summer 2019

Optimization Based Control for Multi-agent System with Interaction

Abstract

Optimization Based Control for Multi-agent System with Interaction

by

Cheng Peng

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

Recently, the artificial intelligence has achieved a significant success with applications in various domains including transportation, smart building, robotics, economy and so on. More and more traditional system entities have been entitled with full or partial autonomy, allowing them to make their own decisions and moves based on the specific surrounding environments. An integration of multiple such intelligent entities is called a multi-agent system (MAS) where the agents need to interact with each other effectively and efficiently to attain cooperation and optimal system performance. As to fulfill this more challenging intelligent interaction objective, the traditional control approaches will not suffice and more advanced algorithms become essential.

In this dissertation, three system structures for interactive control systems, centralized, distributed and decentralized, are discussed with application in intelligent building and autonomous driving. Several concrete interactive control algorithms are proposed and verified.

In the centralized control system, a single central agent with the whole system information available is in charge of making decisions for all the agents. The systemwise cooperation solution is thus directly obtained and all the interactions involved are optimally addressed. Chapter 3 and 4 adopt such centralized control strategy for the intelligent building system. In order to save energy consumption and satisfy the occupants' thermal comfort demand, a combination of feedforward iterative learning control (ILC) and iteratively tuned feedback controller is designed to compensate both repetitive and non-repetitive disturbance components. Chapter 3 proposes an iterative controller design algorithm via optimization solving and stabilizing feedback projection. In Chapter 4, the concurrent design of feedforward ILC and causal stabilizing feedback controller is introduced, where both controllers are simultaneously solved by one optimization.

However, the centralized approach's complexity grows with the problem size, which leads to failure for large-scale systems. The distributed control strategy is introduced as an alternative for such high-dimensional control problems. In the distributed system, a communication network enables the information exchange among agents. Therefore, each agent can keep broadcasting and updating its local controller until a convergence to the cooperative solution is reached. In Chapter 5, a distributed cooperative controller design method is developed for intelligent building thermal control with convergence property theoretically proven.

For a system with no global communication, agents of which follow different control policies, the decentralized control structure is the only valid solution, where each agent designs its local controller independently based on estimated information of others. In Part II of the dissertation, several decentralized interactive control algorithms are proposed for the autonomous driving system. In Chapter 6, an optimization-based negotiation with both concession and persuasion is formulated for vehicle agent's decision making in various interactive scenarios. A Bayesian persuasion based algorithm for interactive driving is explored in Chapter 7. In the algorithm, the ego vehicle agent (persuader) intends to manipulate the interacting vehicle agent (information receiver)'s belief about the current driving situation via observable driving behavior. In Chapter 8, the interaction between two vehicle agents is defined as a two-player persuasion game, the mixed Nash equilibrium of which denotes the agents' optimal intention probabilities. The optimal intention is then expressed via the ego vehicle's driving trajectory planned by an optimization with the intention expression constraint.

To my family

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vi**

1 Introduction **1**
1.1 Background of Multi-agent System with Interacting Agents 1
1.2 Interactive Agent Control 3
1.3 Dissertation Outline 5

I Intelligent Building System **9**

2 Thermal Dynamic System Modeling **10**
2.1 Introduction 10
2.2 Basic Model Structure 12
2.3 Model Identification 13
2.4 Testbed Simulation System Modeling 17
2.5 Model Description in Finite Time Horizon 19
2.6 Chapter Summary 20

3 Centralized Iterative Controller Design **21**
3.1 Introduction 21
3.2 Constrained ILC with Non-repetitive Disturbances 22
3.3 Convex Formulation 24
3.4 Optimal Controller Design 28
3.5 Simulation Results 30
3.6 Chapter Summary 33

4 Centralized Concurrent Controller Design **34**
4.1 Introduction 34
4.2 Optimization Based Concurrent Controller Design 35

4.3 Convexification based on Youla Parametrization 38
4.4 Simulation Results 42
4.5 Chapter Summary 46

5 Distributed Cooperative Controller Design 47
5.1 Introduction 47
5.2 Cooperative ILC 48
5.3 Optimization Problem Formulation 51
5.4 Distributed Cooperative ILC Algorithm and Convergence Analysis 52
5.5 Simulation Results 54
5.6 Chapter Summary 55

II Autonomous Driving System 59

6 Cooperative Driving Based on Negotiation with Persuasion and Concession 60
6.1 Introduction 60
6.2 Cooperative Driving 61
6.3 Conservative Driving Strategy 63
6.4 Human Robot Driver Negotiation 64
6.5 Simulation Results 70
6.6 Chapter Summary 74

7 Bayesian Persuasive Driving 79
7.1 Introduction 79
7.2 General Bayesian Persuasion 81
7.3 Bayesian Persuasive Driving 82
7.4 Simulation Results 91
7.5 Chapter Summary 97

8 Interactive Driving Based on Persuasion Game and Intention Expression 98
8.1 Introduction 98
8.2 Interactive Driving Game 99
8.3 Solution of Nash Equilibria 104
8.4 Trajectory Planning with Intention Expression 107
8.5 Simulation Results 109
8.6 Chapter Summary 114

9 Conclusion and Future Work 116
9.1 Summary 116
9.2 Future Work 118

Bibliography 120

List of Figures

1.1 General structure of an agent in MAS 2
1.2 Structure of an interactive agent in MAS 3

2.1 Heating, ventilation, and air conditioning (HVAC) system 11
2.2 Graphical ARMA model of temperature 14
2.3 Graphical ARMA model of the non-parametric term 16
2.4 Floor plan of the studied room (Room C220, UC Berkeley Cheit Hall) 17
2.5 Train result with disturbance estimation 18
2.6 Test result with disturbance estimation 18
2.7 Simplified HVAC system with four rooms 19

3.1 Current-iteration ILC with iteratively tunable feedback controller 23
3.2 Weather data 30
3.3 Temperature of the four rooms in five iterations 31
3.4 Control input of the four rooms in five iterations 32
3.5 Performance comparison with fixed feedback 32

4.1 Concurrent design of feedforward and feedback controllers 36
4.2 Youla parametrization 40
4.3 Convex concurrent feedforward and feedback controller design 42
4.4 Comparison between disturbance prediction and real disturbance 43
4.5 Concurrent controller design with accurate disturbance prediction 44
4.6 ILC with fixed feedback controller given accurate disturbance prediction 44
4.7 Concurrent controller design with disturbance uncertainties 45
4.8 ILC with fixed feedback controller and disturbance uncertainties 45

5.1 Temperature of the four zones in three iterations 56
5.2 Control input of four zones in three iterations 57
5.3 Algorithm performance in distributed optimization iteration domain 58

6.1 Negotiation with “nice” human driver in lane changing scenario 72
6.2 Negotiation with “tough” human driver in lane changing scenario 73
6.3 Negotiation with “nice” human driver in lane keeping scenario 74

6.4 Negotiation with “tough” human driver in lane keeping scenario 75
6.5 Lane keeping speed profile (negotiation with “nice” human driver case) 75
6.6 Lane keeping speed profile (negotiation with “tough” human driver case) 76
6.7 Negotiation with “nice” human driver in intersection scenario 76
6.8 Negotiation with “tough” human driver in intersection scenario 76
6.9 Intersection crossing speed profile (negotiation with “nice” human driver case) 77
6.10 Intersection crossing speed profile (negotiation with “tough” human driver case) 77

7.1 Bayesian persuasion with “nice” surrounding vehicle in lane changing scenario 92
7.2 Bayesian persuasion with “tough” surrounding vehicle in lane changing scenario . . . 92
7.3 Bayesian persuasion with “nice” surrounding vehicle in lane keeping scenario 93
7.4 Bayesian persuasion with “tough” surrounding vehicle in lane keeping scenario 93
7.5 Lane keeping speed profile (persuasion with “nice” surrounding vehicle case) 94
7.6 Lane keeping speed profile (persuasion with “tough” surrounding vehicle case) 94
7.7 Bayesian persuasion with “nice” surrounding vehicle in intersection scenario 95
7.8 Bayesian persuasion with “tough” surrounding vehicle in intersection scenario 95
7.9 Intersection crossing speed profile (persuasion with “nice” surrounding vehicle case) . 96
7.10 Intersection crossing speed profile (persuasion with “tough” surrounding vehicle case) 96

8.1 Mapping from objective cumulative probability to subjective weight 106
8.2 Game playing with “friendly” surrounding vehicle2 in lane change scenario 110
8.3 Game playing with “aggressive” surrounding vehicle2 in lane change scenario 111
8.4 Lane change speed profile (game with “friendly” surrounding vehicle case) 112
8.5 Lane change speed profile (game with “aggressive” surrounding vehicle case) 112
8.6 Game playing with “friendly” surrounding vehicle in intersection scenario 113
8.7 Intersection crossing speed profile (game with “friendly” surrounding vehicle case) . . 113
8.8 Game playing with “aggressive” surrounding vehicle in intersection scenario 114
8.9 Intersection crossing speed profile (game with “aggressive” surrounding vehicle case) . 114

List of Tables

2.1 Thermal Parameters 19

3.1 Desired temperature when occupied 30

4.1 Desired temperature during working hours 43

6.1 Dimension parameters 71
6.2 Optimization configurations 71

8.1 Game of chicken 100
8.2 Interactive driving game 103
8.3 Interactive driving game event probabilities 106

Acknowledgments

The past fives years at UC Berkeley has been a fruitful and memorable journey in my life. I would not have been where I am today without tremendous help and support from many people.

First and foremost, I would like to express my earnest gratitude to my Ph.D. advisor, Professor Masayoshi Tomizuka, who has completely tranformed my life with his profound knowledge and endless support. He helped shape my research path by giving me the freedom of exploring various challenging and exciting areas and steering me in the right direction when I feel puzzled. Besides the invaluable academic guidance, his insightful lifelong advices, enthusiasm on work and optimistic character has also guaranteed the stability of my "life controller". Without his guidance and persistent help, this dissertation would not have been possible.

I would also like to thank the faculty who helped me during my time at Berkeley. I am thankful to Professor Roberto Horowitz who was my qualifying exam committee chair and served on my dissertation commitee as well. In my first year, I got my first A+ in his ME232 class, which built my confidence and made myself ready for the rest of the Ph.D. journey. I am also grateful to Professor Ruzena Bajcsy and Professor Shmuel S. Oren who served on my qualifying commitee. They not only provided me with insightful comments about research but also lead me to a better understanding of "Ph.D.". Moreover, I would like to thank Professor Anil Aswani for constructive discussion about my dissertation. I also want to thank my undergraduate advisor, Professor Huijun Gao, for introducing me to the world of research.

It is my great fortune to be a member of the Mechanical Systems Control (MSC) laboratory where I gained precious friendship and a lot of research inspirations. Special thanks goes to Professor Wenlong Zhang for his kind instructions and support. I cannot start my research smoothly without his help. I want to thank Shuyang Li, Liting Sun and Wei Zhan for all the brainstorming, valuable discussions and suggestions. I also would like to thank all current and previous members of the MSC lab for their supportive encouragement and always being there for me throughout the years: Chen-Yu Chan, Jianyu Chen, Yujiao Cheng, Yongxiang Fan, Huidong Gao, Kevin Haninger, Yeping Hu, Yu-Chu Huang, Shiyu Jin, Daisuke Kaneishi, Jessica Leu, Jiachen Li, Chung-Yen Lin, Hsien-Chung Lin, Changliu Liu, Junkai Lu, Hengbo Ma, Kiwoo Shin, Lingfeng Sun, Raechel Tan, Chen Tang, Te Tang, Dennis Wai, Changhao Wang, Cong Wang, Taohan Wang, Yizhou Wang, Zining Wang, Zhuo Xu, Xiaowen Yu, Ge Zhang, Yu Zhao, Minghui Zheng, Shiying Zhou, Yiyang Zhou, Xinghao Zhu. I wish you all the best in your future endeavors!

Last but not least, I would like to express my sincerest love to my parents for their unconditioned love, supporting my dreams and making me the person I am now. The final thanks goes to Sirui for every big and small thing that you do for us, sharing joys and sorrows with me over the years. Without you being there for me, I would not have made to this point.

Chapter 1

Introduction

1.1 Background of Multi-agent System with Interacting Agents

Nowadays, with the advancement of control technology, communication system and computer science, the autonomous systems start playing an important role in our daily life. It can be easily imagined that on a normal day in the future, autonomous vehicles will drive people safe and fast to work at an intelligent office building which provides occupants with desired thermal comfort level consuming minimum necessary energy. However, in order to realize such exciting imaginary future, more efforts are required.

Actually, the autonomous driving and intelligent building systems are representative examples of the multi-agent system (MAS). Formally, a MAS is defined as a system with intelligent agents which possess at least partial autonomy. In the autonomous drivng system, all the entities involved in the current driving scenario can be treated as agents incluidng automated vehicles, human-driven vehicles and pedestrians. The agents in the bulding system intuitively correspond to all controllable room/zones. The general structure of an agent in MAS is shown in Fig. 1.1. The agent first obtains sensor data about the environment and other agents. The following node in the information flow is the perception module which analyzes the received sensor data and figures out what the environment looks like. Based on the received world state informaiton produced by the perception module, the planning/high level control module will make decisions telling the agent what action to take. With the next step action decided, the low level controller will then generate corresponding concrete control commands for the agent's actuators.

The MAS technique has been successfully applied in a wide range of domains besides autonomous driving and intelligent building. In the manufacturinng domain, researchers have attempted to solve various problems using the MAS technique such as production planning, resource allocation and process planning for discrete manufacturing operations [35]. The robotics researchers have studied MAS theory in robotics context for several decades. Generally, there are two main challenges for robots control problem: 1) cooperation and coordination and 2) safe and efficient trajectory planning [6]. For both challenges, the MAS approach is able to provide promis-

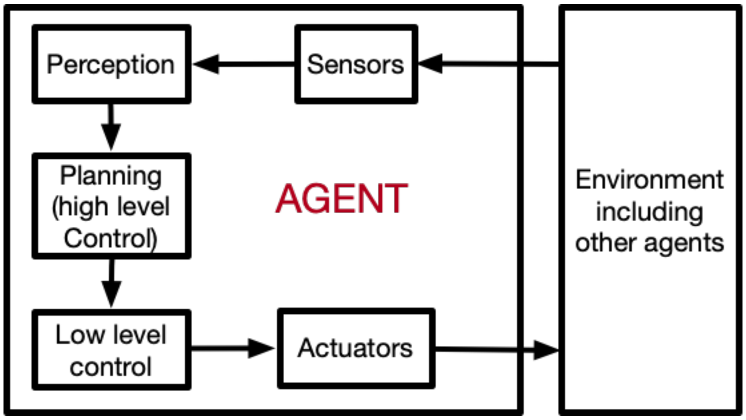
Perception
Sensors
Planning (high level Control)
AGENT
Environment including other agents
Low level control
Actuators

Figure 1.1: General structure of an agent in MAS

ing solutions. For example, in [17], a multi-agent deep reinforcement learning method is proposed for several cooperative multi-agent benchmark tasks; [44] developes a consensus-based algorithm for mobile robots trajectory planning with fixed formation.

Although many of the MAS approach's advantages have been verified by its application popularity, there are still challenges waiting to be addressed. One of the most challenging problems in MAS area is how to handle the interaction. For the automated driving vehicle, first it needs to avoid all the obstacles including pedestrians and other moving or stopping vehicles so that no accidents will happen. Moreover, the interaction is necessary for the scenarios where the vehicles' priorities are not clearly defined, e.g., lane change/keep and intersection crossing in urban driving. In these driving situations, the autonomous vehicle has to "talk" with others and then make its own decision based on the information collected. With regard to the building system, since each room could possibly have different temperature preference, heat interaction between rooms is bound to happen which will not only affect the occupants' thermal comfort but also cause more energy consumption.

In order to deal with the interaction effect in the system, the agent needs to know not only what the environment looks like but also what the environment will look like in the near future. With the prediction module added, the structure of a MAS with interactive agents is shown in Fig. 1.2. Compared to the reflex agent described in Fig. 1.1, after receiving information from the perception module, the interactive agent will first predict about the environment (including other agents)'s future states and utilize the prediction results in the planning/high level control module. Depending on the interactive strategy used, the agent may repeat the prediction-planning procedure for several times to simulate the actual interaction process and the final decision will be passed to the low level control module to make actual moves.

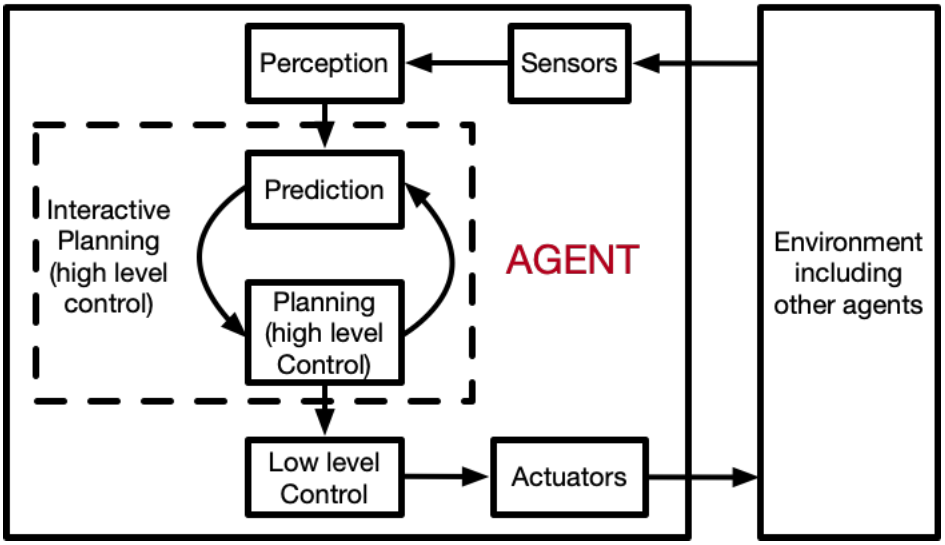
Perception
Sensors
Interactive Planning (high level control)
Prediction
Planning (high level Control)
AGENT
Environment including other agents
Low level Control
Actuators

Figure 1.2: Structure of an interactive agent in MAS

1.2 Interactive Agent Control

In this dissertation, we focus on the interactive planninng (high level control) module design. Generally, there are three different methodologies for the interactive agents control including centralized control, distributed control and decentralized control. The centralized control scheme is mainly applied for small/medium dimensional systems with full information of all the agents available. Since the centralized controller takes all the agents involved in the interaction process into account, the global optimal solution can be achieved. However, the computational load for the centralized approach will keep increasing with the system dimension, which will make the problem intractable for a large-scale system. Additionally, the other agents may not share the same control strategy and the full information may not be accessed. Aiming at resolving such limitations, distributed and decentralized methods have been proposed. For a system with agents using different controllers, if there is a reliable communication system allowing agents to broadcast and receive informaton, the distributed approach can be applied as a solution to the large-scale problem. Otherwise, when the communication is not reliable enough or even not realizable, the decentralized control strategy will be essential.

Centralized Control

The most intuitive interactive control approach is to assign the design task to a single central agent which makes decisions for all the agents. In this way, with all the information available to the central agent, each agent is informed the optimal action so that the interaction effect is dealt in one shot. In order to find the optimal solution, the general centralized control problem can be formulated as an optimization:

$$\begin{aligned} \min_{c_1,\cdots,c_n} \quad & J(c_1,\cdots,c_n) \\ s.t. \quad & \text{constraints}, \end{aligned} \tag{1.1}$$

where c_1 to c_n represent the controllers corresponding to each agent and J denotes the global cost function for the whole system depending on all the agents' controllers.

For most real engineering systems, such centralized control structure's application is quite limited due to its heavy dependency on the number of agents in the system and the full information requirement. However, for a building system with a fair number of control zones, the centralized approach apperas to be the right solution. Usually, a small/medium-size biulding system is only equipped with one building management system which controls all zones in the building meaning that all the related innformation is accessible. Such organization structure is naturally in line with the centralized control.

Distributed Control

With regard to the large-scale system with communication between agents, the distributed approach can be applied to reduce the computational load. A distributed control system is composed of local controllers associated with each agent and a communication system through which the agents are able to exchange decision information with each other. Via the informatoin exchange process, the decisions made by local controllers will converge to the problem's optimum which guarantees the system control performance similar with the centralized approach. The distributed optimization-based control problem for the agent i can be formulated as:

$$\begin{aligned} \min_{c_i} \quad & J(\hat{c}_1,\cdots,c_i,\cdots,\hat{c}_n) \\ s.t. \quad & \text{constraints}, \end{aligned} \tag{1.2}$$

where c_i is the decision-making agent i's local controller, J is the same global cost function as in (1.1) and $\hat{c}$ represents the controller result information broadcasted by other agents. In the application, each agent will solve such local optimization in parallel and broadcast the obtained solutions to the whole agent network. Based on the updated information of the other agents (understanding of the whole system), a new controller will be designed accordingly for each agent. After several information exchange iterations, the local controller solutions will converge to an optimum which mean the cooperation among agents after interaction.

The success of the distributed control algorithm is rooted in the reliable communication system. If the information exchange process is corrupted, the agents' prediction or understanding of the whole system will be deviated from the real situation. Such deviation will lead to failure of convergence and degrade the control performance. For the building application, as all zones are physically connected, such communication system can be fairly easy to build. Therefore, the distributed control becomes a valid solution to the large-scale intelligent building control problem.

Decentralized Control

For a system where each agent follows a unique control strategy and the reliable global communication system does not exist, the decentralized control method will be the only solution. In a decentralized system, agents actions are determined by independent local control systems with different strategies and goals. Therefore, in order to handle the interaction effect and resolve possible conflicts, the agents have to talk with others in some manner, building a virtual communication system. In this virtual communincation system, the agent should convey to others the information it intends to express via behavior so that the other agents can adjust their future decisions accordingly. Meanwhile, the agent also needs to update its belief of its interacting agents and make judgement about their possible future actions after observing their new behavior. Basically, this process can be seen as a negotiation between agents, which can be done in various concrete forms dependent on the system characteristics.

Mathematically, the decentralized optimal control design can be fulfilled via the following decentralized optimization:

$$\begin{aligned} \min\nolimits_{c_i} \quad & \tilde{J}_i(\tilde{c}_1, \cdots, c_i, \cdots, \tilde{c}_n) \\ s.t. \quad & \text{constraints}, \end{aligned} \tag{1.3}$$

where c_i represents agent i's controller to be designed, $\tilde{J}_i$ is the global cost function of the system estimated by agent i and $\tilde{c}$'s denote the estimated/predicted control decisions of other agents in the system. Due to the lack of exact information about the whole system, to find a balance between agents' different demands usually requires multiple rounds of negotiation and optimization.

The autonomous drivng system is a typical example of the decentralized control structure application since different vehicles/drivers seldom share the exactly same driving strategy and there exists many barriers for real-time communication during driving process. Therefore, how to design the negotiation mechanisms for agents becomes the key challenge in the autonomous and interactive driving.

1.3 Dissertation Outline

In this dissertation, several methodologies are developed for control of interactive agents in two applications: intelligent building system and autonomous driving system. Three major control structures for multi-agent systems are discussed: centralized, distributed and decentralized. The centralized approach is applied to small/medium sized builidng system control; the distributed structure is utilized to solve the large-scale challenge for high dimennsional building system and the decentralized control strategy is explored for the autonomous driving system. The dissertation is organized as follows.

Intelligent Building System

In Part I of the dissertation, the temperature regulation problem for intelligent building system is studied. In Chapter 2, a model identification approach is introduced to parametrize the building thermal dynamic model. Since the building system is essentially an integrated entity with all zones connected together, centralized control approaches are first introduced in Chapter 3 and Chapter 4 to reject disturbances and guarantee satisfactory control performance. For the large-scale building system, a distributed cooperative control algorithm is developed in Chapter 5.

Thermal Dynamic System Modeling

The first step for the controller design is modeling and identification of the plant. In Chapter 2, in order to facilitate the controller design, the complex thermal dynamic system is modeled as a first-order lumped resistance and capacitance (RC) model. Since such simplified model structure cannot capture all the complex thermal effects, a semiparametric regression estimation algorithm is applied. This statistical techinique represents all the uncertain and complex effects by a non-parametric term and therefore can achieve an accurate estimation for the parametric part. With the identified model, a model description in finite time horizon using lifting technique [4] is developed which allows us to design controllers via time-domain optimization.

Centralized Iterative Controller Design

Although the centralized control strucutre offers an optimal strategy to handle interaction effects, more efforts are required to achieve other control objectives for the thermal system including robustness against disturbance, energy saving and comfort level expected by the occupants. In Chapter 3, an iterative design of feedforward and feedback controller is proposed. The algorithm aims at finding a balance between building occupants' thermal comfort and the energy consumption. The influence of disturbance from ambient environment and inside human activities is considered. Observing that the disturbances are nearly repetitive, an iterative learning control (ILC) strategy is utilized to eliminate the influence of repetitive disturbance and an iteratively tuned feedback controller is applied to reduce the influence of non-repetitive disturbance components. With constraints from input saturation, the controller design is formulated into a constrained optimization problem. To reduce the complexity of the problem, an alternating procedure is proposed to solve the optimization problem, separating the original problem into two parts, namely, constrained ILC problem and iterative feedback tuning problem. The proposed algorithm is demonstrated by simulations on a four-room testbed system [42], [43].

Centralized Concurrent Controller Design

The iterative controller design approach requires solving multiple optimization problems and the resulting controller is an approximation of the optimal feedforward and feedback combination. To find the exact optimum of the problem, in Chapter 4, a concurrent design of feedforward and feedback controllers is proposed. The controller design is formulated as an optimization with feedforward ILC and Youla-parametrized feedback controller as two blocks for decision making. To guarantee that the controllers are stabilizing and implementable in practice, stability, causality and robustness requirements must be considered along with the control saturation limit as constraints. The optimization is convexified via introducing new variables and relaxation. The solution to the optimization simultaneously gives the feedforward and feedback controllers to be applied in the next iteration. The proposed algorithm is demonstrated on a four-room system by simulation [40].

Distributed Cooperative Controller Design

For a large-scale building system, the centralized control structure will suffer from high problem dimension, making the control design extremely time-consuming or even intractable. For such large-scale systems, a distributed and cooperative ILC algorithm is proposed in Chapter 5. For each agent, a distributed optimization is formulated for the ILC design, the cost function of which is global and shared by all the agents. After solving the distributed optimization, each agent will broadcast the obtained solutions to its interacting neighbors and receive the neighbors' updated controller solutions at the same time. Via such informaiton exchange process, all the agents' local solutions will converge to a global cooperative optimum. The convergence property of the algorithm is theoretically proven and simulation results are provided to demonstrate the algorithm's effectiveness [38].

Autonomous Driving System

Part II of this dissertation is devoted to the autonomous driving system considering interaction between individual vehicles. In Chapter 6, the interaction between vehicle agents is formulated as an optimization-based negotiation with concession and persuasion. Chapter 7 introduces a Bayesian persuasion algorithm for agent decision making in interactive driving scenarios. In Chapter 8, the interactive driving problem is explicitly desgined as a two-player persuasion game.

Cooperative Driving Based on Negotiation with Persuasion and Concession

In the future, to build an intelligent driving environment, it is necessary for the autonomous vehicles to effectively and efficiently interact with other road users including human-driven vehicles and other autonomous vehicles. However, due to the lack of a global and reliable communication system, the agents do not have access to the others' intention states and decision making strategies and cannot directly convey their intentions to others either. In order to achieve cooperation and successful interaction without information transparency, the agent has to negotiate with others about right of the road via observable driving behavior. In Chapter 6, such negotiation process with concession and persuasion is formulated as an optimization. Concession is introduced as a tuning weight in the cost function and persuasion is formulated as a convex constraint. The proposed algorithm enables the autonomous vehicle to find a balance between conservativeness and aggressiveness so that the interaction process can be finished smoothly [41].

Bayesian Persuasive Driving

Since the interactive driving problem is still a unsolved open challenge, there exist many possible solutions. In Chapter 7, another decentralized interactive control approach based on Bayesian persuasion is proposed. In the persuasion process, the ego vehicle intends to convey its intention and future decision information to the interacting vehicle via its driving behavior. The information received by the interacting vehicle will impose certain influence on its belief about the current driving environment, which will potentially affect its next step behavior decision making. The proposed algorithm aims at finding the optimal driving behavior which will persuade the interacting vehicle to alter its behavior towards a better global welfare [39].

Interactive Driving Based on Persuasion Game and Intention Expression

Essentially, the interactive decision making process is a game between the interacting agents. The game's Nash equilibrium solution represents the best strategy for the vehicles in the current situation. In Chapter 8, a two-player persuasion game is formulated and the mixed Nash equilibrium is found. In the game setting, each player/agent has two options representing the vehicles' intention which are yielding and passing. The game's payoff matrix is dependent on both vehicles' driving style information estimated based on Bayesian update. In order to help agents make more human-like decisions, the cumulative prospect theory (CPT) is applied mapping the payoff matrix to a utility function. The Nash equilibrium with the optimal yielding/passing (intention) probabilities is then solved corresponding to the developed utility. In the trajectory design module, the obtained intention probability is formulated as a convex constraint imposed on the vehicle speed. In this manner, the optimal intention expressive trajectory for the ego autonomous vehicle is found and the intelligent interaction is achieved.

Part I

Intelligent Building System

Chapter 2

Thermal Dynamic System Modeling

In general, modeling serves as the foundation of controller design and is thus critical to the system performacne. With an accurate well-structured model, the designed controller will potentially reject disturbances effectively and guarantee a desired control performance. In this chapter, a parametrized model for the thermal dynamics is introduced and the model parameters are estimated using semiparametric approach.

2.1 Introduction

Currently, the building sector consumes about 20-40% of the worldwide energy and exceeds consumption of other major sectors, including industry and transportation. Moreover, it is forecast that the energy cost in buildings will keep growing by 34% in the next two decades. This striking phenomenon is due to the growth of population worldwide, increasing requirement for comfort levels and more time spent inside the building. Majority of this building energy consumption is caused by the heating, ventilation, and air conditioning (HVAC) system, which consumes approximately 50% of it. Therefore, in order to balance between people's growing demand for thermal comfort and energy saving, more advanced HVAC control strategies become necessary [45]. Among many key factors for controller design, an accurate model is the most important and fundamental one. Recently, many different HVAC models have been proposed through simulation software. However, such computational models developed are always high-dimensional and too complicated for controller design. A trade-off has to be made between the model complexity and model accuracy.

Figure 2.1 shows the general structure of a HVAC system. The most important sector in the HVAC system is the air handling unit (AHU), which determines the maixmum air mass flow rate into the system and the supply air temperature. The input of an AHU is the air flow mixed by the outside air and return air from the system, which is then heated or cooled to the determined supply air temperature by the heating and cooling coils. The output air with desired temperature is forced into the duct system by a supply fan. Before entering each room, the supply air flow will go through dampers which adjust mass flow rate of air by changing the ducts' opening areas. At last, part of air is recirculated and mixed with outside air in order to go through the system again.

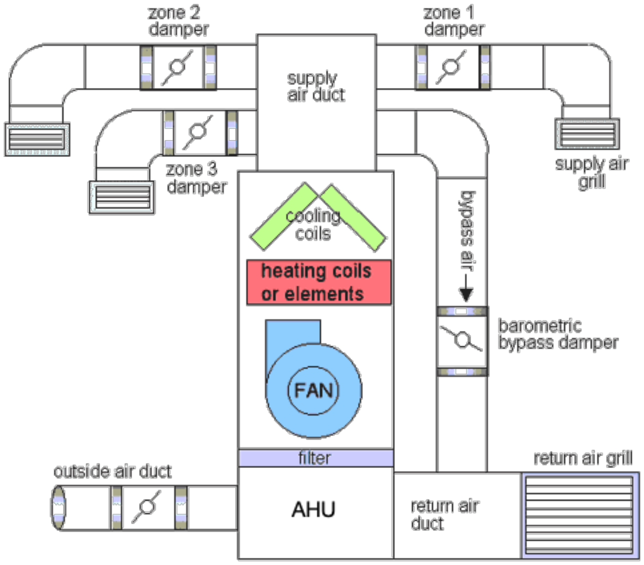

Figure 2.1: Heating, ventilation, and air conditioning (HVAC) system

In order to properly describe the thermal dynamic system functioning in the context of HVAC application, a lumped first-order resistance and capacitance (RC) model [5] is utilized in this dissertation which parametrizes the building system with thermal resistances and capacitances. The model parameters involved can either be theoretically derived based on thermodynamics knowledge or estimated from measurement data. However, the thermodyncmics derivation approach usually requires a full understanding of all the building materials' complex thermal properties. Moreover, there is not a general derivation approach valid for all different structured buildings. Therefore, in this dissertation, a data-driven model estimation approach is applied to get a numerical description of the HVAC thermal dynamic system.

However, in the real building system, there exist certain factors which cannot be fully described by a parametric model. Some examples include the measurement error in the data record, undetected disturbance sources and thermal effects beyond the scope of RC model. In order to achieve an estimated model with good performance in presence of such uncertainties, a semiparametric model combining both parametric part and non-parametric part is applied. The parametric part reflects the thermal RC influence and all the uncertain factors are merged together as a non-parametric term in the model. To achieve numerical values of the parameters, an autoregressive-moving average (ARMA) time series model is developed and the maximum loglikelihood algorithm is used to find the optimal solution. For the prediction and verificatoin purpose, another ARMA model is developed for the non-parametric part and the corresponding parameters are achieved via the expectation-maximization (EM) algorithm. With the obtained model parameters, a model description in the finite time horizon based on the lifted system technique is introduced to facilitate optimization-based controller design.

The remainder of this chapter is organized as follows. The basic thermal model structure is introduced in Section 2.2. Section 2.3 proposes a semiparametric model parameters estimation approach based on ARMA model and a specific case is studied demonstating the approach's effectivenss. Section 2.4 applies the model estimation method to a four-room testbed system and Section 2.5 introduces a lifted model description friendly to time-domain control design. Section 2.6 concludes the chapter.

2.2 Basic Model Structure

Although specific thermal system models differ from case to case, they still share some general structures. Therefore, instead of treating the thermal dynamic system as a complete black box, we treat it as a grey box with known structure and several unknown parameters. According to the heat transfer principles, a first order thermal RC model can be achieved to reflect the thermal dynamic evolution relationship [5]. The most obvious benefit of this model is its simplicity and computational efficiency. This model uses concepts of thermal capacitance and resistance, analogous to the electric capacitance and resistance. The basic assumptions utilized include:

- The air inside each zone is well-mixed, which means that the whole zone has the uniform temperature. Thus each zone can be treated as a simple temperature node.
- Thermal properties of the room studied is constant during the control process, which indicates that all the resistances and capacitances in the RC thermal network are constant.
- The density of air is constant, implying that net mass of air in the room is unchanged.

These assumptions are actually valid for most commercial and non-commercial buildings.

Using the assumptions above, the HVAC temperature model can be formulated as

$$c_i \frac{dT_i}{dt} = \sum_k^n \frac{1}{R_{ki}}(T_k - T_i) + Q_i, \tag{2.1}$$

where c_i and T_i are the thermal capacitance and temperature of zone i respectively, T_k's represent the temperature of zone i's neighboring zones, including the ambient environment, R_{ki} represents the thermal resistance of heat transfer between zone i and zone k and Q_i denotes the heat flux into zone i. By decomposing Q_i into two parts including the energy control input and disturbances, the RC model in (2.1) can be rewritten as

$$\frac{dT_i}{dt} = c_i^{-1} \sum_k^n \frac{1}{R_{ki}}(T_k - T_i) + c_i^{-1} u_i + c_i^{-1} v_i, \tag{2.2}$$

where v_i denotes the heat disturbance of zone i and

$$u_i = \dot{m} c_p (T_c - T_i)$$

is zone i's energy control input, determined by $\dot{m}_i$: the air mass flow rate to zone i, T_c: the supply air temperature and c_p: the specific heat capacity of air.

After discretization with a sampling time of T_s, the following discrete time model is obtained:

$$\begin{aligned} T_i(t+1) &= T_i(t) + T_s(c_i^{-1}\sum_k^n \frac{1}{R_{ki}}(T_k(t) - T_i(t)) + c_i^{-1}u_i(t) + c_i^{-1}v_i(t)) \\ &= (1 - T_s c_i^{-1}\sum_k^n \frac{1}{R_{ki}})T_i(t) + T_s c_i^{-1}\sum_k^n \frac{1}{R_{ki}}T_k(t) + c_i^{-1}u_i(t) + c_i^{-1}v_i(t) \end{aligned}$$

To simplify the expression above, two new variables are introduced: $a_i = c_i^{-1}$ and $b_{i,k} = c_i^{-1}\frac{1}{R_{ki}}$ which reformulate the model as

$$T_i(t+1) = (1 - T_s\sum_k^n b_{i,k})T_i(t) + T_s\sum_k^n b_{i,k}T_k(t) + a_i u_i(t) + a_i v_i(t). \tag{2.3}$$

2.3 Model Identification

Semiparametric Method

In the building systems, the disturbance mainly come from two sources, which are the inside human activities and outside weather. The baseline of the disturbances can be obtained from the room's working schedule and the weather forecast respectively. However, the real situation can be far more complex, there will be many unexpected uncertainties corrupting the prediction results. For instance, the schedule for room usaeg is subject to change as some meetings may be cancelled or rescheduled. With regard to the ambient environment, since the mechanism of weather system has not been fully understood, the forecast information cannot be perfectly accurate. All these possibilities make the real influence of disturbance hard to predict, which may deteriorate fitting performance of the learned model. Hence, in this dissertation, a semiparametric method is utilized to address such uncertainty issue.

In statistics, a semiparametric model [3] is a model with both parametric and nonparametric components. When applied in the model estimation problem, the basic idea of the semiparametric method is also very straightforward: merging all the unknown factors into a new nonparametric variable and performing system model identification for the parametric part. In the thermal dynamics context, a new variable $q_i(t)$ is introduced to represent all the thermal uncertainties for room i. Thus, the model can be separated into two parts and reformulated as

$$T_i(t+1) = (1 - T_s\sum_k^n b_{i,k})T_i(t) + T_s\sum_k^n b_{i,k}T_k(t) + a_i u_i(t) + a_i w_i(t) + q_i(t) + \varepsilon(t), \tag{2.4}$$

where $w_i(t)$ represents the predicted disturbance to zone i and $\varepsilon(t)$ is assumed to be independent and identically distributed zero mean noise with constant covariance.

Now the question becomes how to learn the parametric part (a_i's and $b_{i,k}$'s) from the collected data. First, the following conditional expectations are defined: $\hat{T}_i(t) = \mathbb{E}[T_i(t)|t]$, $\hat{u}_i(t) = \mathbb{E}[u_i(t)|t]$, $\hat{T}_k(t) = \mathbb{E}[T_k(t)|t]$, $\hat{w}_i(t) = \mathbb{E}[w_i(t)|t] = w_i(t)$ and $\hat{q}_i(t) = \mathbb{E}[q_i(t)|t] = q_i(t)$. With these notations, by taking conditional expectations for both sides of (2.4) and using the fact that $\mathbb{E}[\varepsilon(t)] = 0$, the following equation can be obtained:

$$\hat{T}_i(t+1) = (1 - T_s\sum_k^n b_{i,k})\hat{T}_i(t) + T_s\sum_k^n b_{i,k}\hat{T}_k(t) + a_i\hat{u}_i(t) + a_i w_i(t) + q_i(t). \tag{2.5}$$

Substracting (2.5) from (2.4) achieves

$$T_i(t+1) - \hat{T}_i(t+1) = (1 - T_s\sum_k^n b_{i,k})(T_i(t) - \hat{T}_i(t)) + T_s\sum_k^n b_{i,k}(T_k(t) - \hat{T}_k(t)) + a_i(u_i(t) - \hat{u}_i(t)), \tag{2.6}$$

where the non-parametric term $q_i(t)$ is eliminated and a pure parametric model is left. To obtain values of $\hat{T}_i$, $\hat{T}_k$ and $\hat{u}_i$, we just need to smooth data of T_i, T_k and u_i over time. There are many different tools that can fulfill this task, in this dissertation, the kernel regression technique is applied.

Mixture of ARMA Time-Series Model

In this section, the time-series thermal model behavior is described by the ARMA structure [53] and the optimal estimation of parameters a_i and $b_{i,k}$ are obtained. The basic idea of ARMA model is to predict future states using the past information in time series for a stationary stochastic process. With the model structure already determined by the RC approach, the numbers of autoregressive terms and moving average terms are both selected to be 1. To build a clear model description, several notations are introduced: $D = (D_1, \cdots, D_t, \cdots, D_N)$ where $D_t = T_i(t) - \hat{T}_i(t)$, $D_k = (D_{k,1}, \cdots, D_{k,t}, \cdots, D_{k,N})$ where $D_{k,t} = T_k(t) - \hat{T}_k(t)$ and $D_u = (D_{u,1}, \cdots, D_{u,t}, \cdots, D_{u,N})$ where $D_{u,t} = u_i(t) - \hat{u}_i(t)$. Using these notations, the temperature ARMA model can be represented by a directed graph as shown in Fig. 2.2, the mathematical description of which is

$$D_{t+1} : \mathscr{N}(\mu_{D,t+1}, \sigma^2), \tag{2.7}$$

where $\mu_{D,t+1} = (1 - T_s\sum_k^n b_{i,k})D_t + T_s\sum_k^n b_{i,k}D_{k,t} + a_i D_{u,t}$ and σ is set to be 1.

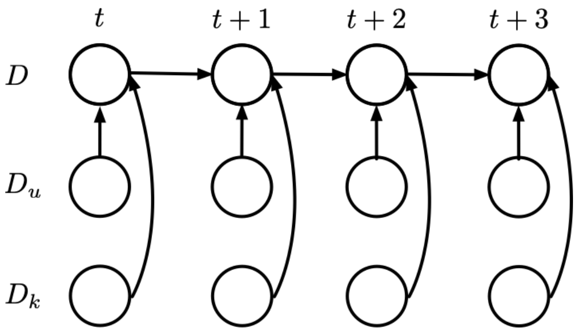

Figure 2.2: Graphical ARMA model of temperature

Denoting all the parameters by $\theta = [b_{i,1}, \cdots, b_{i,n}, a_i]^T$, the loglikelihood can be calculated as

$$\begin{aligned} l(\theta) &= log(\prod_t \frac{1}{\sqrt{2\pi}} exp(-\frac{(D_t - \mu_t)^2}{2})) \\ &= -\frac{1}{2}\sum_t (D_t - \mu_t)^2 + log(2\pi). \end{aligned}$$

By applying the maximum loglikelihood estimation algorithm, the following equations can be obtained:

$$\begin{aligned} \frac{\partial l}{\partial b_{i,k}} &= \sum_{t=2}^{N} (D_t - \mu_t)(T_s D_{k,t-1} - T_s D_{t-1}) = 0, \\ \frac{\partial l}{\partial a_i} &= \sum_{t=2}^{N} (D_t - \mu_t)(T_s D_{u,t-1}) = 0. \end{aligned} \tag{2.8}$$

To find solutions to (2.8), the Newton-Raphson method is applied with the update law as:

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1} \nabla_\theta (l(\theta^{(t)})), \tag{2.9}$$

where H is the Hessian of loglikelihood l and $\nabla_\theta(l)$ is the gradient of l. The hessian H can be calculated as

$$H = \begin{bmatrix} Bb & Bc \\ * & Cc \end{bmatrix}, \tag{2.10}$$

where Bb is an $n \times n$ matrix with the elements

$$Bb_{ij} = -\sum_{t=2}^{N} (T_s D_{i,t-1} - T_s D_{t-1}) * (T_s D_{j,t-1} - T_s D_{t-1}), i, j = 1, \cdots, n,$$

Bc is an $n \times 1$ vector with elements

$$Bc_i = \sum_{t=2}^{N} T_s D_{u,t-1} (T_s D_{i,t-1} - T_s D_{t-1}), i = 1, \cdots, n,$$

and

$$Cc = -\sum_{t=2}^{N} T_s^2 D_{u,t-1}.$$

By repeating (2.9) for several iterations, the estimation of parameters a_i and $b_{i,k}$ can be achieved.

In order to verify the estimated model's effectiveness, prediction results for the testing data are required. However, at this stage, the model still cannot be used for temperature prediction as the non-parametric part is totally unknown. Therefore, we propose to train another ARMA model for the non-parametric term q_i. The time series training data for q_i can be extracted from the original data set using the estimated a_i and $b_{i,k}$:

$$\hat{q}_i(t) = \hat{T}_i(t+1) - (1 - T_s \sum_k^n b_{i,k}) \hat{T}_i(t) - T_s \sum_k^n b_{i,k} \hat{T}_k(t) - a_i \hat{u}_i(t) - a_i w_i(t), \tag{2.11}$$

where $\hat{q}_i(t)$ represents the estimation of $q_i(t)$ used as training data. Actually, as q_i represents the disturbance uncertainties caused by inside human activities and outside weather, it is reasonable to assume that q_i at the current time step depends on its own past values. which validates the application of ARMA model method. Similar with the thermal parameters estimation procedure, several notations are introduced including $Q = (Q_1, \cdots, Q_t, \cdots, Q_N)$ where $Q_t = q_i(t)$ and latent white noise variables $E = (E_1, \cdots, E_N)$. The graphical description of the ARMA model for q_i is shown in Fig. 2.3, the mathematical description of which is as:

$$Q_{t+1} : \mathscr{N}(\mu_{Q,t+1}, \sigma_q^2), \tag{2.12}$$

where $\mu_{Q,t} = \psi + \sum_{k=1}^{r} \alpha_k Q_{t-k} + E_{t-1}$, r is the autoregressive order, ψ and α's are model parameters to be estimated and σ_q is again set to be 1.

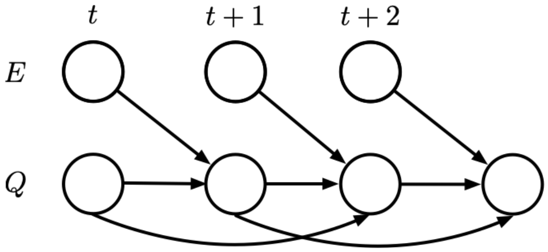

Figure 2.3: Graphical ARMA model of the non-parametric term

Denoting all the parameters as $\theta_q = [\psi, \alpha_1, \cdots, \alpha_r]^T$ and utilizing EM algorithm, the estimation process of ARMA model for the non-parametric term is as

- E-step:

$$\begin{aligned} l(\theta_q) &= \mathbb{E}\left[log(\prod_{t=h}^{N} \frac{1}{\sqrt{2\pi}} exp(-\frac{(Q_t - \mu_{Q,t})^2}{2}))\right] \\ &= -\frac{1}{2}\sum_{t=h}^{N} \mathbb{E}[(Q_t - \mu_{Q,t})^2] + log(2\pi). \\ &= -\frac{1}{2}\sum_{t=h}^{N} \mathbb{E}[(Q_t - (\psi + \sum_{k=1}^{r} \alpha_k Q_{t-k} + E_{t-1}))^2] + log(2\pi), \end{aligned}$$

  where $h = \max(N, r+1)$.

- M-step:

$$\psi^{j+1} = \frac{\sum_{t=h}^{N} \hat{q}_i(t) - \sum_{k=1}^{r} \alpha_k^j \hat{q}_i(t-k)}{N-h+1} \tag{2.13}$$

$$\alpha_k^{j+1} = \frac{\sum_{t=h}^{N} \Big(\hat{q}_i(t-k)\hat{q}_i(t) - \psi^j \hat{q}_i(t-k) - \hat{q}(t-k)\sum_{l=1,l\neq k}^{r} \alpha_l^j \hat{q}(t-l)\Big)}{\sum_{t=h}^{N} \hat{q}_i(t-k)}, \tag{2.14}$$

  where j denotes the EM iteration.

After running the EM algorithm for several iterations, the convergence can be achieved resulting in well-trained ARMA model for the non-parametric term.

Case Study Results

In this section, a classroom at UC Berkeley is used as a case study and the temperature prediction results are shown to demonstrate the introduced model identification method's effectiveness. The temperature measurements and working (class) schedule are collected from room C220 of UC Berkeley Cheit Hall, the floor plan of which is shown in Fig. 2.4. The temperature data of the studied room from Oct. 21st to Oct. 29th is used for training and the data from Oct. 31st to Nov. 17th is used for testing. The sampling time of the system is 15 minutes and the autoregressive order of the non-parametric part's ARMA model is set as 2. The training and testing results are shown in Fig. 2.5 and Fig. 2.6 respectively, where the good prediction performance is achieved indicating an accurate estimated model.

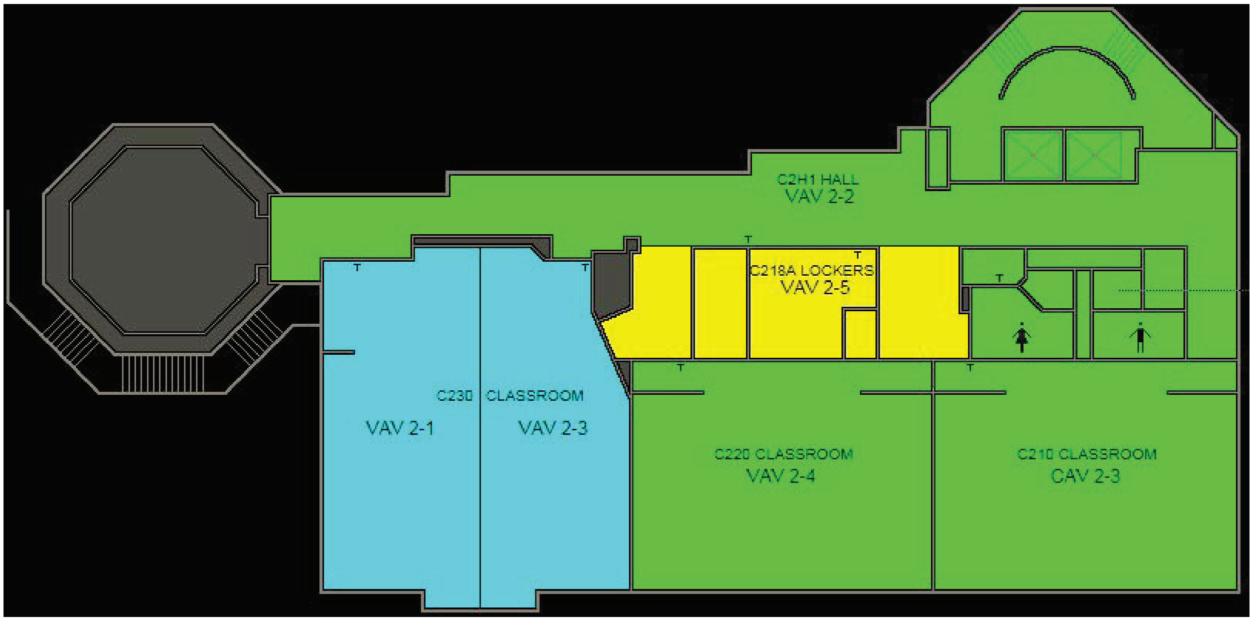

Figure 2.4: Floor plan of the studied room (Room C220, UC Berkeley Cheit Hall)

2.4 Testbed Simulation System Modeling

Unfortunately, the Chiet Hall room cannot be used as a testing system for the controller design purpose due to the safety reasons. Therefore, in this section, a simulation system model is developed based on a four-room testbed system built in Singapore, the layout of which is shown in Fig. 2.7.

In the testbed system, there is one single air handling unit (AHU) serving all the four rooms and each room has its own controllable damper. Following the same model structure defined in

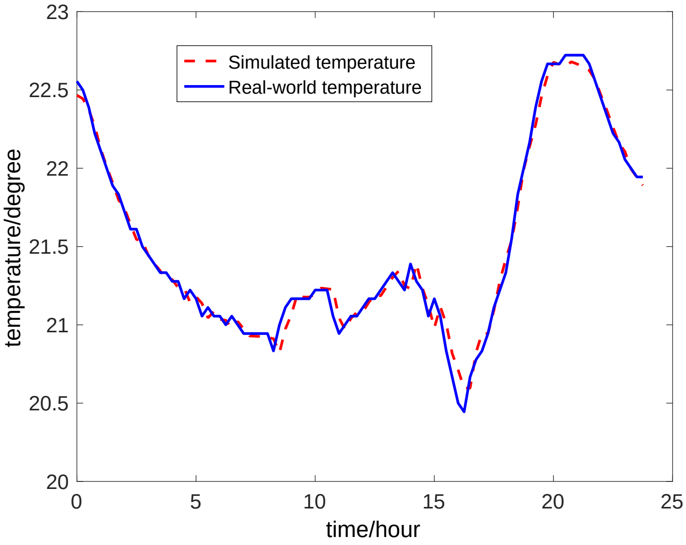

Figure 2.5: Train result with disturbance estimation

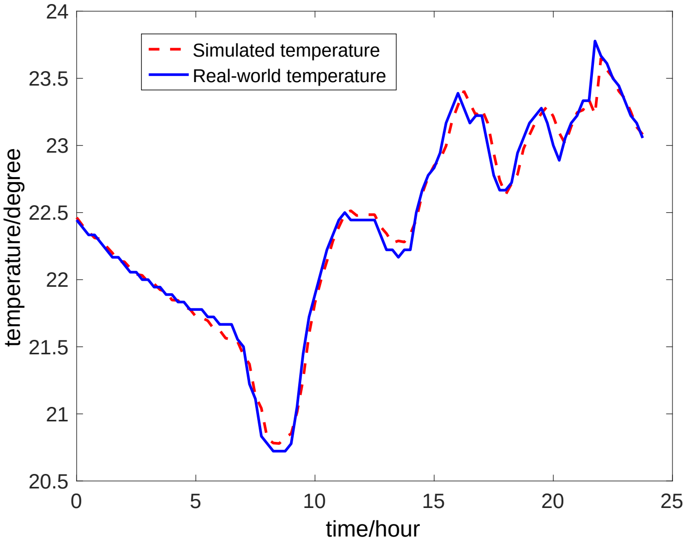

Figure 2.6: Test result with disturbance estimation

(2.1) and combining all the four rooms, the centralized system model can be derived as:

$$C_T\frac{dT}{dt} = R_T T + R_{T0}T_0 + u + w, \tag{2.15}$$

where $T=[T_1,\cdots,T_4]^T$ and $u=[u_1,\cdots,u_4]^T$ represent the temperature vector and the energy input (control input) vector for the 4 zones respectively. T_0 denotes the ambient temperature, and $w=[w_1,\cdots,w_4]^T$ is the disturbance vector for all zones including effect of the non-parametric term and inside thermal activities. $C_T \in \mathbb{R}^{4\times4}$ and $R_T \in \mathbb{R}^{4\times4}$ represent the thermal capacitance and

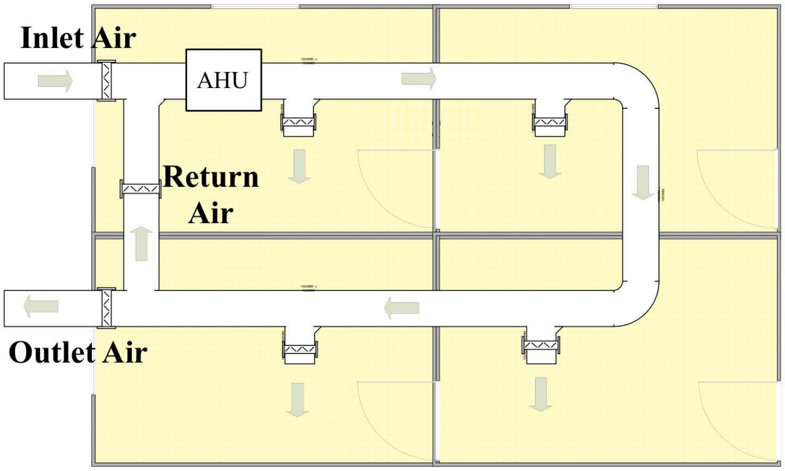

Figure 2.7: Simplified HVAC system with four rooms

Table 2.1: Thermal Parameters

Parameter	Value	Unit
c_1, c_2, c_3, c_4	1000	kJ/K
$R_{12}, R_{13}, R_{24}, R_{34}$	1.0417	kW/K
R_{14}, R_{23}	10000	kW/K
$R_{01}, R_{02}, R_{03}, R_{04}$	0.5208	kW/K

resistance matrix respectively, describing the thermal interaction among all zones. $R_{T0} \in \mathbb{R}^{4\times 1}$ is a thermal resistance vector corresponding to the system's thermal interaction with the outside environment. The corresponding discretized state space model of system (2.15) can be developed as:

$$G : T(t+1) = A_g T(t) + B_g u(t) + B_g d(t), \tag{2.16}$$

where $A_g = I + T_s C_T^{-1} R_T$, $B_g = T_s C_T^{-1}$ are system matrices and $d(t) = w(t) + R_{T0} T_0(t)$ is the lumped disturbance vector. The parameter matrices C_T, R_T and R_{T0} can be estimated with the semiparametric model identification approach described in this chapter using real world measurement data. The specific thermal parameters used in this dissertation are shown in Table 2.1.

2.5 Model Description in Finite Time Horizon

In this section, the developed thermal dynamic model is described using the lifted form within finite time horizon so that optimization-based control design approach can be applied.

To design controllers in time domain with finite time horizon, an intuitive idea is to stack all the signals into a super vector. Such a technique is called lifting [4], which has been widely used for time-domain optimization based control problems.

First define a general single-input-single-output (SISO) discrete dynamics system as

$$y(t) = h(z^{-1})u(t), \tag{2.17}$$

where $y(t)$ and $u(t)$ are the output and input signals, respectively.

$$h(z^{-1}) = h_1 z^{-1} + h_2 z^{-2} + \cdots, \tag{2.18}$$

is the system's transfer function written in the infinite impulse response form, where the parameters $[h_0, h_1, \cdots]$ are the system's impulse response at each time instant.

Then, the output signals in a finite time horizon N can be expressed in the vector and matrix form:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} h_1 & 0 & \cdots & 0 \\ h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ h_N & h_{N-1} & \cdots & h_1 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix},$$

which can be compactly written as

$$\mathbf{Y}{=}\mathbf{HU}. \tag{2.19}$$

The system (2.19) is called the lifted presentation of the system (2.17). The super vector **Y** obtained by stacking all the signals in the time horizon is defined to be the original signal y's lifted form and the matrix **H** is defined as the system's impulse resposne matrix.

For the remainder of Part I, signals/systems in lifted form are in bold form and the corresponding original signals/systems are in normal form.

Note that the lifting technique is also applicable to multiple-input-multiple-output (MIMO) systems. The only difference is that in MIMO system's impulse response matrix **H**, each h element is a matrix instead of a scalar. Therefore, the building thermal dynamics system (2.16) can be lifted as:

$$\mathbf{T} = \mathbf{G}(\mathbf{U} + \mathbf{D}) + \mathbf{T}^{in}, \tag{2.20}$$

where **T**, **U**, **D** and $\mathbf{T}^{in}$ represent the lifted form of temperature T, control input u, disturbance d and the system's initial temperature state respectively. Matrix **G** represents the lifted system plant.

2.6 Chapter Summary

In this chapter, the general HVAC system structure is introduced and a lumped first-order resistance and capacitance (RC) model is given to describe the building thermal dynamics. In order to achieve accurate model parameters for both prediction and control purpose, the model is identified using a semiparametric approach, where two ARMA time-series models are trained for the parametric and non-parametric components respectively. The identification approach's effectiveness is verified by a case study. For the controller design purpose, a simulation testbed system model is developed and its description in finite time horizon based on lifted system is discussed.

Chapter 3

Centralized Iterative Controller Design

With the model developed in Chapter 2, a centralized iterative controller design approach is proposed in this chapter. The designed control system is composed of a constrained iterative learning control (ILC) and an iteratively tunable feedback controller. To guarantee good control performance in the presence of both repetitive and non-repetitive disturbances, the feedforwrad ILC and the feedback controller are designed based on optimization in each iteration. Considering constraints from the input saturation, the ILC convergence requirement and the closed-loop stability, the controller design is formulated as a convex optimization problem. The influence of disturbance uncertainties is also incorporated into the optimization problem in the form of chance constraints. To reduce the complexity of the problem, techniques such as relaxation and projection on convex sets are introduced to make the algorithm more efficient.

3.1 Introduction

Among many alternatives to traditional building control approaches, model predictive control (MPC) has been extensively investigated and applied. It aims at minimizing the energy consumption over a planning time window subject to certain constraints. Weather forecast information can be incorporated to handle the influence of disturbance [25], [30]. However, in practice, neither the weather forecast nor the estimation of human activities can be exact. Such disturbance uncertainties may severely influence the MPC performance.

Note that even though disturbances from weather and human activities cannot be precisely known, there always exist certain repetitive patterns for both of them, especially for office and commercial buildings. Considering such repetitiveness in disturbance, iterative learning control (ILC) may be effective for building control [33]. The central idea of ILC is to improve the performance of current iteration based on learned information from previous iterations. Therefore, the repetitive disturbance can be effectively eliminated by ILC even if it is unknown. Another difference of ILC compared to MPC is that ILC design process is offline, which means that computation load will not be a significant issue.

In the ILC design for the building temperature control systems, the two most critical issues are the effect of non-repetitive disturbance and the input saturation constraint. Both of them, if not properly addressed, can severely degrade the control performance. With regard to the effect of non-repetitive disturbance, several frequency-domain ILC design schemes have been proposed [51], [27]. However, in such frequency-domain approaches, it is hard to incorporate constraints. On the other hand, in [50], [7], [16], several optimization-based approaches are proposed to solve constrained ILC problems, but non-repetitive disturbance is not taken into consideration.

In this chapter, an optimization-based ILC with iterative tunable feedback controller is proposed to handle both the constraints and the non-repetitive disturbances. By translating the input saturation limit, the ILC convergence and the closed-loop stability requirements into convex constraints, the control problem is formulated as an optimization problem. The feedforward ILC input is used to handle the repetitive disturbances and the iteratively optimized feedback controller is introduced to deal with the non-repetitive components. Moreover, a chance constraint method is incorporated to further enhance the performance robustness against disturbance uncertainties.

The remainder of this chapter is organized as follows. The control design problem is defined in section 3.2 and section 3.3 formulates the problem into a solvable convex optimization problem. Section 3.4 describes the implementation of the algorithm. Simulation results are given in section 3.5 to demonstrate the effectiveness of the proposed algorithm. Section 3.6 concludes the chapter.

3.2 Constrained ILC with Non-repetitive Disturbances

The fundamental idea of ILC is to learn from past, attempting to improve the performance of current iteration based on experience from previous iterations. Through learning, ILC can effectively remove the unknown repetitive disturbances. In addition, benefiting from its offline property, ILC will not bring significant computation load for online application. To better adapt ILC into the HVAC control problem, there are still two remaining issues: influence of the non-repetitive disturbances and constraints. In this section, the ILC design is formulated as a constrained optimization and a feedback controller is used to handle the non-repetitive disturbances.

The ILC update strategy adopted in this dissertation is called current-iteration ILC [4]. Utilizing the lifted technique described in section 2.5, the ILC update law is given by

$$\begin{array}{c} \mathbf{U}_{k+1} = \mathbf{U}_{ff,k+1} + \mathbf{C}_{k+1}\mathbf{E}_{k+1} \\ \mathbf{U}_{ff,k+1} = \mathbf{U}_k + L_{k+1}\mathbf{E}_k \end{array}, \tag{3.1}$$

where k, $\mathbf{U}_{ff}$, $\mathbf{E}$, L, $\mathbf{U}$ and $\mathbf{C}$ represent the iteration index, the feedforward ILC input, the error signal, the learning filter, the control input to the plant and the feedback controller respectively.

Obviously, the performance of ILC heavily depends on the learning filter L. Traditionally, such a learning filter can be analytically given by minimizing a quadratic cost function, with constant feedback controller and no constraints. However, in the building temperature control problem, certain constraints have to be incorporated and the existence of non-repetitive disturbance calls for an iteration-varying feedback controller. To address these problems, an ILC algorithm with iteration-tunable feedback controller is defined as follows.

The block diagram of the proposed control system is shown in Fig. 3.1. To formulate the

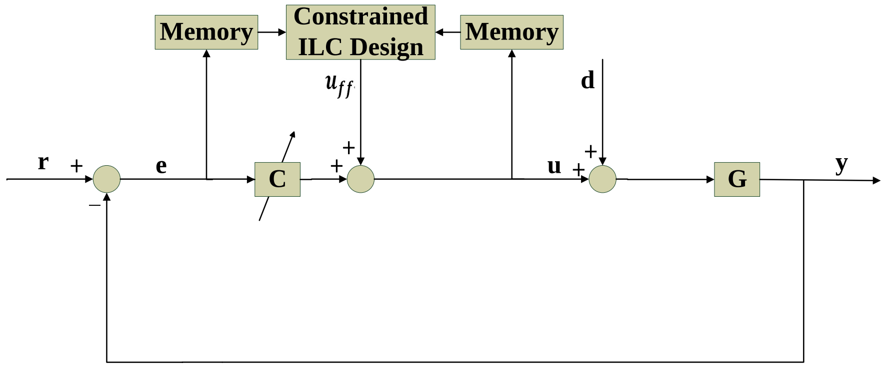

Figure 3.1: Current-iteration ILC with iteratively tunable feedback controller

problem into an optimization problem, a quadratic cost function in iteration domain is defined as

$$J_{k+1} = \mathbf{E}_{k+1}^T W_e \mathbf{E}_{k+1} + \mathbf{U}_{k+1}^T W_u \mathbf{U}_{k+1}, \tag{3.2}$$

where W_e and W_u are positive definite penalty matrices for error signal $\mathbf{E}$ and control input $\mathbf{U}$, respectively. The variables in this optimization problem include $\mathbf{C}_{k+1}$ and L_{k+1}. Utilizing the lifted system technique and the relationship in (3.1), the control signal $\mathbf{U}_{k+1}$ in Fig. 3.1 is given by

$$\mathbf{U}_{k+1} = \mathbf{T}_{uff2u,k+1}(\mathbf{U}_k + L_{k+1}\mathbf{E}_k) + \mathbf{T}_{d2u,k+1}\mathbf{D}_{k+1} + \mathbf{T}_{r2u,k+1}\mathbf{R} \tag{3.3}$$

where

$$\mathbf{T}_{uff2u,k+1} = (I + \mathbf{C}_{k+1}\mathbf{G})^{-1},$$

$$\mathbf{T}_{d2u,k+1} = -(I + \mathbf{C}_{k+1}\mathbf{G})^{-1}\mathbf{C}_{k+1}\mathbf{G},$$

and

$$\mathbf{T}_{r2u,k+1} = (I + \mathbf{C}_{k+1}\mathbf{G})^{-1}\mathbf{C}_{k+1}$$

represent the lifted transfer functions from $\mathbf{U}_{ff,k+1}$ to $\mathbf{U}_{k+1}$, from disturbance $\mathbf{D}_{k+1}$ to $\mathbf{U}_{k+1}$ and from reference signal $\mathbf{R}$ to $\mathbf{U}_{k+1}$, respectively. $\mathbf{G}$ represents the lifted system plant. Similarly, the error signal is given by

$$\mathbf{E}_{k+1} = \mathbf{T}_{r2e,k+1}\mathbf{R} + \mathbf{T}_{uff2e,k+1}(\mathbf{U}_k + L_{k+1}\mathbf{E}_k) + \mathbf{T}_{d2e,k+1}\mathbf{D}_{k+1}, \tag{3.4}$$

where

$$\mathbf{T}_{r2e,k+1} = (I + \mathbf{G}\mathbf{C}_{k+1})^{-1},$$

$$\mathbf{T}_{uff2e,k+1} = -(I + \mathbf{G}\mathbf{C}_{k+1})^{-1}\mathbf{G},$$

and

$$\mathbf{T}_{d2e,k+1} = -(I + \mathbf{G}\mathbf{C}_{k+1})^{-1}\mathbf{G}$$

are the lifted transfer functions from reference $\mathbf{R}$ to error $\mathbf{E}_{k+1}$, from feedforward control input $\mathbf{U}_{ff,k+1}$ to $\mathbf{E}_{k+1}$ and from disturbance $\mathbf{D}_{k+1}$ to $\mathbf{E}_{k+1}$.

Therefore, high nonlinearity exists in the minimization problem of J_{k+1} as the variables $\mathbf{C}_{k+1}$ and L_{k+1} are coupled with each other in $\mathbf{E}_{k+1}$ and $\mathbf{U}_{k+1}$. Moreover, $\mathbf{C}_{k+1}$ appears in the inverse form in all the lifted transfer function terms.

Meanwhile, several constraints need to be included. In HVAC system, one of the most important constraints is the saturation constraint on the input signal U, which is given by

$$\underline{U} \leq \mathbf{U}_{k+1} \leq \overline{U}, \tag{3.5}$$

where $\underline{U}$ is the lower bound of $\mathbf{U}$ and $\overline{U}$ is the upper bound of $\mathbf{U}$. Substituting (3.3) into (3.5), the saturation constraint can be expressed in the lifted form as

$$\underline{U} \leq \mathbf{T}_{uff2u,k+1}\left(\mathbf{U}_k + L_{k+1}\mathbf{E}_k\right) + \mathbf{T}_{d2u,k+1}\mathbf{D}_{k+1} + \mathbf{T}_{r2u,k+1}\mathbf{R} \leq \overline{U}. \tag{3.6}$$

Besides, new constraints from closed-loop stability arise as the feedback controller C is varying between iterations. Suppose the closed-loop system is realized by $[A_c, B_c, C_c, D_c]$, then the closed-loop stability requires that all the eigenvalues of A_c are inside the unit circle, namely,

$$|\lambda(A_c)| \leq 1. \tag{3.7}$$

Moreover, monotonic convergence of ILC has to be guaranteed for good control performance. Thus, another ILC convergence constraint is included as

$$||\mathbf{E}_{k+1}||_2 \leq ||\mathbf{E}_k||_2, \tag{3.8}$$

which guarantees the decrease of the error signal in iteration domain (in the sense of 2-norm).

Summarizing (3.2-3.8), the ILC design problem has been defined as a constrained optimization problem:

$$\begin{array}{cl} \min_{L_{k+1},\mathbf{C}_{k+1}} & J_{k+1} \\ s.t. & |\lambda(A_c)| \leq 1 \\ & \underline{U} \leq \mathbf{U}_{k+1} \leq \overline{U} \\ & ||\mathbf{E}_{k+1}||_2 \leq ||\mathbf{E}_k||_2. \end{array} \tag{3.9}$$

The problem in (3.9) is, however, not convex due to the high nonlinearity in both the cost function and constraints. To make it solvable, special techniques such as relaxation and projection on convex sets are employed in this chapter and will be discussed in the following sections.

3.3 Convex Formulation

In this section, the defined nonconvex optimization problem in (3.9) will be reformulated as a solvable convex optimization problem.

Introduction of New Variables

Recall that the control signal U_{k+1} in (3.3) and the error signal E_{k+1} in (3.4) are both nonlinear functions with respect to variables C_{k+1} and L_{k+1}. To get rid of the nonlinearity, two new variables X and Y are defined as

$$\begin{aligned} Y &= (I+\mathbf{C}_{k+1}\mathbf{G})^{-1}, && (3.10)\\ X &= (I+\mathbf{C}_{k+1}\mathbf{G})^{-1}L_{k+1}. && (3.11) \end{aligned}$$

Then the control input signal in (3.3) can be rewritten as:

$$\begin{aligned} \mathbf{U}_{k+1} &= (I+\mathbf{C}_{k+1}\mathbf{G})^{-1}(\mathbf{U}_{ff,k+1}+\mathbf{C}_{k+1}\mathbf{R}-\mathbf{C}_{k+1}\mathbf{G}\mathbf{D}_{k+1})\\ &= (I+\mathbf{C}_{k+1}\mathbf{G})^{-1}(\mathbf{U}_k+L_{k+1}\mathbf{E}_k)\\ &\quad +[I-(I+\mathbf{C}_{k+1}\mathbf{G})^{-1}](\mathbf{G}^{-1}\mathbf{R}-\mathbf{D}_{k+1})\\ &= Y\mathbf{U}_k+X\mathbf{E}_k+(I-Y)\mathbf{G}^{-1}\mathbf{R}-(I-Y)\mathbf{D}_{k+1}. \qquad (3.12) \end{aligned}$$

Noting the fact that for any matrices $A\in\mathbb{R}^{n\times m}$ and $B\in\mathbb{R}^{m\times n}$,

$$(I_n+AB)^{-1}A = A(I_m+BA)^{-1} \qquad (3.13)$$

always holds, the error signal in (3.4) can be simplified as

$$\begin{aligned} \mathbf{E}_{k+1} &= (I+\mathbf{G}\mathbf{C}_{k+1})^{-1}(\mathbf{R}-\mathbf{G}\mathbf{U}_{ff,k+1}-\mathbf{G}\mathbf{D}_{k+1})\\ &= (I+\mathbf{G}\mathbf{C}_{k+1})^{-1}\mathbf{G}\mathbf{G}^{-1}\mathbf{R}\\ &\quad -(I+\mathbf{G}\mathbf{C}_{k+1})^{-1}\mathbf{G}(\mathbf{U}_k+L_{k+1}\mathbf{E}_k+\mathbf{D}_{k+1})\\ &= \mathbf{G}(I+\mathbf{C}_{k+1}\mathbf{G})^{-1}\mathbf{G}^{-1}\mathbf{R}\\ &\quad -\mathbf{G}(I+\mathbf{C}_{k+1}\mathbf{G})^{-1}(\mathbf{U}_k+L_{k+1}\mathbf{E}_k+\mathbf{D}_{k+1})\\ &= \mathbf{G}Y\mathbf{G}^{-1}\mathbf{R}-\mathbf{G}Y\mathbf{U}_k-\mathbf{G}X\mathbf{E}_k-\mathbf{G}Y\mathbf{D}_{k+1}. \qquad (3.14) \end{aligned}$$

Through such mathematical manipulations, both (3.12) and (3.14) now have become linear functions of variables Y and X. Consequently, substituting (3.12) and (3.14) into (3.2) and replacing unavailable future disturbance $\mathbf{D}_{k+1}$ with predicted disturbance $\hat{\mathbf{D}}_{k+1}$ yields the objective function as

$$\begin{aligned} J_{k+1} &= \|W_e^{\frac{1}{2}}[\mathbf{G}Y(\mathbf{G}^{-1}\mathbf{R}-\hat{\mathbf{D}}_{k+1})-\mathbf{G}(Y\mathbf{U}_k+X\mathbf{E}_k)]\|_2^2\\ &\quad +\|W_u^{\frac{1}{2}}[Y\mathbf{U}_k+X\mathbf{E}_k+(I-Y)(\mathbf{G}^{-1}\mathbf{R}-\hat{\mathbf{D}}_{k+1})]\|_2^2, \end{aligned} \qquad (3.15)$$

which is jointly convex with respect to variables X and Y.

The introduction of X and Y also allows simplification of the input saturation constraint as

$$\underline{U} \leq Y\mathbf{U}_k+X\mathbf{E}_k+(I-Y)\mathbf{G}^{-1}\mathbf{R}-(I-Y)\hat{\mathbf{D}}_{k+1} \leq \overline{U}. \qquad (3.16)$$

Similarly, the ILC convergence constraint in (3.8) can be simplified as

$$\|\mathbf{G}Y\mathbf{G}^{-1}\mathbf{R} - \mathbf{G}Y\mathbf{U}_k - \mathbf{G}X\mathbf{E}_k - \mathbf{G}Y\hat{\mathbf{D}}_{k+1}\|_2 \leq \|\mathbf{E}_k\|_2. \tag{3.17}$$

Note that when designing the controller for the $(k+1)$-th iteration, the error signal $\mathbf{E}_k$ in the k-th iteration, is already known, so the ILC convergence constraint in (3.17) is a standard conic constraint in vector norm form and ready to solve. To further enhance the performance robustness against disturbance uncertainties, additional chance constraints are introduced in the next section.

Chance Constraint

As mentioned before, in practice, weather forecast is inevitably subject to uncertainties, which means that the weather forecast can only provide the baseline information. This is also true for the human activity disturbance prediction. Thus, blind trust of disturbance prediction may violate the saturation constraint in (3.16) and correspondingly leads to poor control performance.

To incorporate the influence of disturbance uncertainties into the optimization problem, a chance constraint approach [37] is proposed, which reformulates the original input saturation constraint in (3.16) to

$$\mathbf{Prob}[\underline{U} \leq Y\mathbf{U}_k + X\mathbf{E}_k + (I-Y)\mathbf{G}^{-1}\mathbf{R} - (I-Y)\mathbf{D}_{k+1} \leq \overline{U}] \geq \eta, \tag{3.18}$$

where η is the confidence level. Assume that the uncertain disturbance is Gaussian,

$$\mathbf{D}_{k+1} \sim \mathscr{N}(\hat{\mathbf{D}}_{k+1}, \Sigma), \tag{3.19}$$

where Σ is the covariance matrix and can be selected based on historical disturbance information. Denote the i-th element of the vector $Y\mathbf{U}_k + X\mathbf{E}_k + (I-Y)\mathbf{G}^{-1}\mathbf{R}$ as b_i and the i-th row of the matrix $I-Y$ as a_i^T, then the control input will also be Gaussian and its distribution can be expressed as

$$b_i - a_i^T\mathbf{D}_{k+1} \sim \mathscr{N}(b_i - a_i^T\hat{\mathbf{D}}_{k+1}, a_i^T\Sigma a_i), i = 1, \cdots, n. \tag{3.20}$$

Hence, the chance constraint (3.18) can be rewritten as:

$$\mathbf{Prob}(b_i - a_i^T\mathbf{D}_{k+1} \leq \overline{U}) = \Phi\left(\frac{\overline{U} - b_i + a_i^T\hat{\mathbf{D}}_{k+1}}{\sqrt{a_i^T\Sigma a_i}}\right) \geq \eta,$$

and

$$\mathbf{Prob}(b_i - a_i^T\mathbf{D}_{k+1} \geq \underline{U}) = \Phi\left(\frac{b_i - a_i^T\hat{\mathbf{D}}_{k+1} - \underline{U}}{\sqrt{a_i^T\Sigma a_i}}\right) \geq \eta,$$

$i = 1, \cdots, n$, which are equal to:

$$\begin{array}{l} \overline{U} - b_i + a_i^T\hat{\mathbf{D}}_{k+1} \geq \Phi^{-1}(\eta)\|\Sigma^{\frac{1}{2}}a_i\|_2 \\ b_i - a_i^T\hat{\mathbf{D}}_{k+1} - \underline{U} \geq \Phi^{-1}(\eta)\|\Sigma^{\frac{1}{2}}a_i\|_2 \end{array}, i = 1, \cdots, n, \tag{3.21}$$

where Φ is the standard Gaussian cumulative distribution function.

Closed-loop Stability Constraint

Since the feedback system is tuned iteratively, it is necessary to guarantee the closed-loop stability in each iteration.

In this chapter, the feedback controller is employed with $u_{fb,k}(t) = K_k e_k(t)$, where e_k and K_k represent the error signal and the feedback gain in the k-th iteration respectively. Recalling the system dynamics in (2.16), when ignoring the disturbance, the closed-loop system in the k-th iteration can be formulated as

$$T_k(t+1) = (A_g - B_g K_k) T_k(t) + B_g K_k r(t) \tag{3.22}$$

where $r(t)$ is the reference signal. Denote $A_{c,k} = A_g - B_g K_k$, then the stability constraint requires that all the eigenvalues of the non-symmetric matrix $A_{c,k}$ be within the unit circle. Generally, optimization problems subjected to eigenvalue constraints are very difficult to solve, especially for high dimensional case. Therefore, a Lyapunov approach [11] is applied to formulate this constraint into the solvable convex linear matrix inequality (LMI) form.

Theorem I: The closed-loop system realized by $[A_c, B_c, C_c, D_c]$ is stable if there exist full rank matrices F, M and $S = S^T > 0$ with appropriate dimension such that

$$\begin{bmatrix} F^T + F - S & F^T A_g^T - M^T B_g^T \\ A_g F - B_g M & I \end{bmatrix} > 0, \tag{3.23}$$

and the feedback gain K can be recovered as $K = MF^{-1}$.

Proof : According to the Lyapunov stability theorem for discrete time system, the closed-loop system realized by $[A_c, B_c, C_c, D_c]$ is stable if and only if there exists a symmetric positive definite matrix Q such that

$$A_c^T Q A_c - Q < 0. \tag{3.24}$$

Using the Schur complement and substituting in $A_c = A_p - B_p K$, the result in (3.24) can be shown to be equivalent to

$$\begin{bmatrix} Q & (A_g - B_g K)^T \\ (A_g - B_g K) & Q^{-1} \end{bmatrix} > 0. \tag{3.25}$$

The inequality in (3.25) is a bilinear matrix inequality (BMI), which is still hard to solve. Thus, to transfer it to a LMI form, first left-multiply and right-multiply (3.25) by $\begin{bmatrix} F^T & 0 \\ 0 & I \end{bmatrix}$ and $\begin{bmatrix} F & 0 \\ 0 & I \end{bmatrix}$, where F is a full rank real matrix with appropriate dimension. This multiplication gives

$$\begin{bmatrix} F^T Q^{-1} F & F^T (A_g - B_g K)^T \\ (A_g - B_g K) F & I \end{bmatrix} > 0. \tag{3.26}$$

Then using the inequality $F^T + F - Q^{-1} \leq F^T Q^{-1} F$, a sufficient condition of (3.26) can be achieved:

$$\begin{bmatrix} F^T + F - Q^{-1} & F^T (A_g - B_g K)^T \\ (A_g - B_g K) F & I \end{bmatrix} > 0. \tag{3.27}$$

Since there still exists inversion and product of design variables in (3.27), it cannot be efficiently solved. Introduce two new variables $M = KF$ and $S = Q^{-1}$, then the LMI in (3.23) can be obtained. The feedback gain K can be recovered as $K = MF^{-1}$. (End of proof)

Applying Theorem I to the ILC design problem combining ILC convergence constraint in (3.17) and chance constraint of input saturation in (3.21), the controller design problem can be restated as

$$\min_{X,Y,F,M,S} J_{k+1} \tag{3.28}$$

subject to

$$\begin{array}{c} \left.\begin{array}{l} \overline{U} - b_i + a_i^T \hat{\mathbf{D}}_{k+1} \geq \Phi^{-1}(\eta) \|\Sigma^{\frac{1}{2}} a_i\|_2 \\ b_i - a_i^T \hat{\mathbf{D}}_{k+1} - \underline{U} \geq \Phi^{-1}(\eta) \|\Sigma^{\frac{1}{2}} a_i\|_2 \end{array}\right., i = 1, \cdots, n, \\ \|\mathbf{G}Y\mathbf{G}^{-1}\mathbf{R} - \mathbf{G}Y\mathbf{U}_k - \mathbf{G}X\mathbf{E}_k - \mathbf{G}Y\hat{\mathbf{D}}_{k+1}\|_2 \leq \|\mathbf{E}_k\|_2, \\ \begin{bmatrix} F^T + F - S & F^T A_p^T - M^T B_g^T \\ A_g F - B_g M & I \end{bmatrix} > 0. \end{array}$$

3.4 Optimal Controller Design

As a 2-DOF (degree of freedom) optimization-based controller design approach, the optimization problem in (3.28) involves variables in lifted form and non-lifted form. To make these variables reconcile in one framework, a three-step optimization method is proposed. In the first step, the optimization problem is solved in the lifted form, namely, minimizing J_{k+1} subject to input saturation constraint and ILC convergence constraint. Recalling (3.15), (3.17) and (3.21), the problem of this initializing solution step can be defined as

$$\min_{X,Y} J_{k+1} \tag{3.29}$$

subject to

$$\begin{array}{c} \left.\begin{array}{l} \overline{U} - b_i + a_i^T \hat{\mathbf{D}}_{k+1} \geq \Phi^{-1}(\eta) \|\Sigma^{\frac{1}{2}} a_i\|_2 \\ b_i - a_i^T \hat{\mathbf{D}}_{k+1} - \underline{U} \geq \Phi^{-1}(\eta) \|\Sigma^{\frac{1}{2}} a_i\|_2 \end{array}\right., i = 1, \cdots, n, \\ \|\mathbf{G}Y\mathbf{G}^{-1}\mathbf{R} - \mathbf{G}Y\mathbf{U}_k - \mathbf{G}X\mathbf{E}_k - \mathbf{G}Y\hat{\mathbf{D}}_{k+1}\|_2 \leq \|\mathbf{E}_k\|_2. \end{array}$$

The optimal solution (X^*, Y^*) gives the optimal learning matrix L_{k+1}^* and optimal feedback controller $\mathbf{C}_{k+1}^*$ by

$$L_{k+1}^* = Y^{*\dagger} X^* \tag{3.30}$$

$$\mathbf{C}_{k+1}^* = \left(Y^{*\dagger} - I\right) \mathbf{G}^{-1} \tag{3.31}$$

where $(\cdot)^\dagger$ denotes the pseudo inverse of a matrix.

However, such optimal $\mathbf{C}_{k+1}^*$ is not necessarily causal to be implemented. Therefore, in the second step, the optimal feedback controller $\mathbf{C}_{k+1}^*$ from the first step is projected to an implementable causal stabilizing feedback controller for the closed-loop stability. In this stabilizing projection step, a new optimization problem is defined as

$$\min_{F,S,M} \left\| \mathbf{M}\hat{\mathbf{E}}^*_{k+1} - \mathbf{C}^*_{k+1}\mathbf{F}\hat{\mathbf{E}}^*_{k+1} \right\|_2, \tag{3.32}$$

subject to

$$\begin{bmatrix} F^T + F - S & F^T A_g^T - M^T B_g^T \\ A_g F - B_g M & I \end{bmatrix} > 0,$$

$$S > 0,$$

where $\hat{\mathbf{E}}^*_{k+1}$ is the estimated error signal for the $(k+1)$-th iteration with the optimal learning matrix L^*_{k+1} and controller $\mathbf{C}^*_{k+1}$. $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{F} \in \mathbb{R}^{n\times n}$ are block diagonal matrices with block entries M and F respectively. The goal of the objective function in (3.32) is to find the closest stabilizing controller to $\mathbf{C}^*_{k+1}$ with minimum difference in control performance.

The problem in (3.32) is a standard semi-definite problem (SDP). Note that as $\mathbf{M}$ annd $\mathbf{F}$ in the objective function are just the lifted form of M and F, so real design variables only include F, S, and M. Once the optimal solution is found as (M^*, S^*, F^*), the feedback gain K^*_{k+1} can be given as

$$K^*_{k+1} = M^* F^{*\dagger}. \tag{3.33}$$

With K^*_{k+1}, the initial solution L^*_{k+1} can be updated for better performance. This leads to a simplified process of the first initializing solution step, with the $\mathbf{C}_{k+1}$ fixed and replaced by $\mathbf{K}_{k+1}$, which is the lifted form of K^*_{k+1}.

Overall, the proposed algorithm can be summarized as:

- Step 1: Initializing solution. Solve the optimization problem in (3.29) to obtain an initial solution of $\mathbf{C}_{k+1}$ and L_{k+1} satisfying the input saturation constraint and the ILC convergence constraint.

- Step 2: Stabilizing projection. Solve the optimization problem in (3.32) to project the non-causal $\mathbf{C}^*_{k+1}$ to a stabilizing feedback controller K^*_{k+1}.

- Step 3: Update. Solve the optimization problem in (3.29) for a updated learning filter L'_{k+1} with $\mathbf{C}_{k+1}$ fixed and replaced by $\mathbf{K}_{k+1}$.

Remark 1: Note that the proposed algorithm cannot theoretically guarantee the feasibility, so if infeasibility happens, the control input obtained from the last iteration can be applied instead. In fact, the infeasibility issue can only happen under extreme weather condition, meaning that the outside environment temperature is unusually high or low compared to the previous iterations. However, this harsh condition may not actually happen for most building systems.

3.5 Simulation Results

Simulation Setup

The simulation was conducted on the testbed system shown in Fig. 2.7 with thermal parameters in Table 2.1. During working hours from 8am to 7pm, the desired temperatures for each room are as shown in Table 3.1. Otherwise, the reference temperatures are set to be 30 degrees. With regard to the disturbance, the two-day weather forecast data and real weather data in Singapore, collected from EnergyPlus energy simulation software website [14] are shown in Fig. 3.2. It is seen that the disturbances have highly non-repetitive components and significant uncertainties exist. Note that in the simulation, weather forecast data and real weather data for five consecutive days are used.

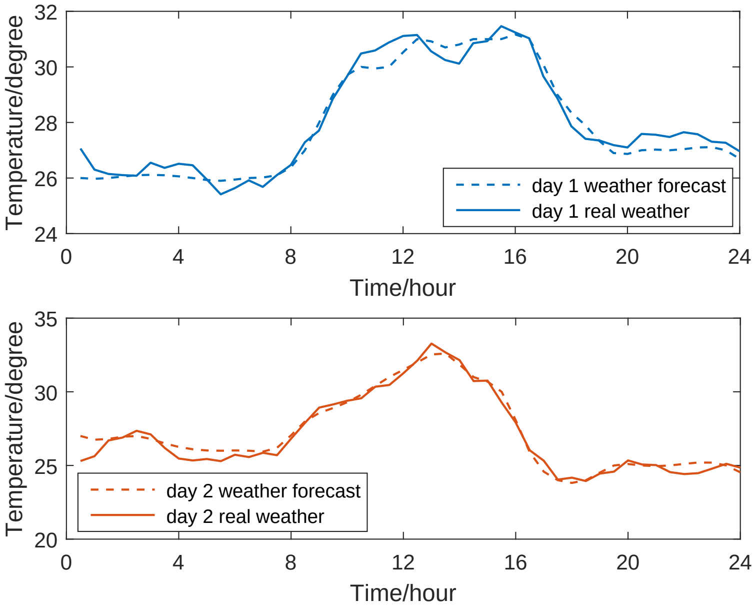

Figure 3.2: Weather data

Table 3.1: Desired temperature when occupied

Room	1	2	3	4
Temperature(°C)	24	25	26	27

Discussion of Results

In this section, simulation results are shown to demonstrate the effectiveness of the proposed algorithm. The weather forecast data is used as disturbance prediction information $\hat{D}$ when designing

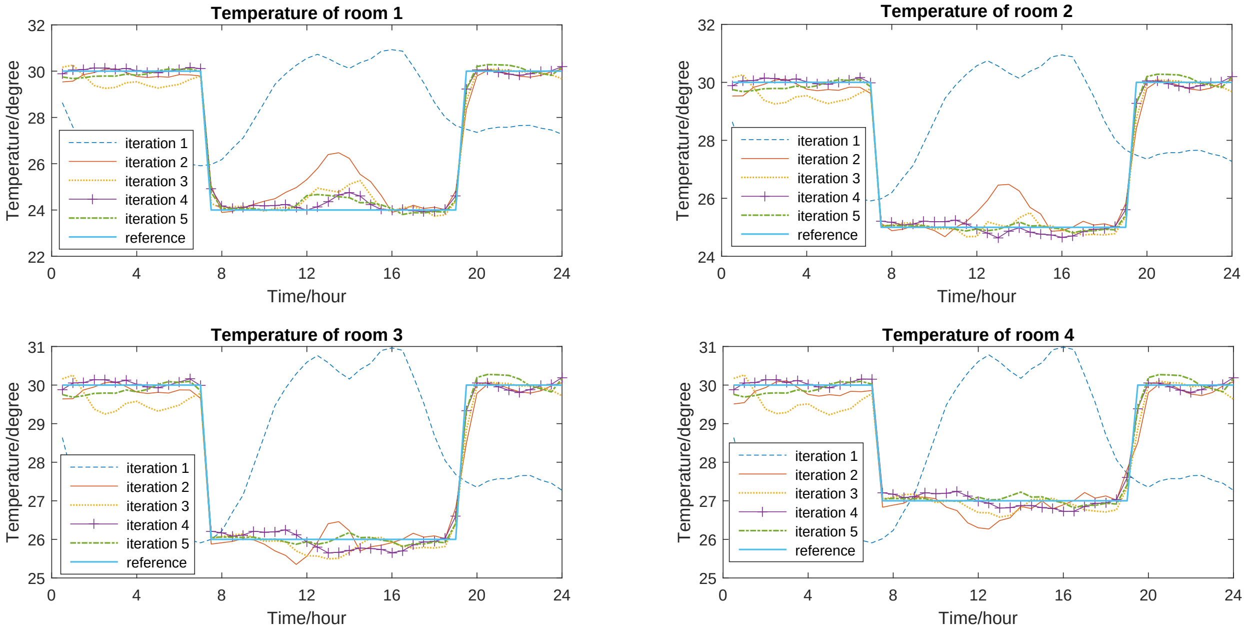

Figure 3.3: Temperature of the four rooms in five iterations

controllers and real weather data is applied to the system as real disturbance D in simulation. The simulation was run on the laptop lenovo yoga 900 with Matlab 2015b. The upper bound of the control input is set to be 2 and the lower bound is set to be -2. The ILC and feedback controller are updated simultaneously once every day, meaning that one iteration is one-day period.

The temperature profiles of the four rooms from the first iteration to the fifth iteration are shown in Fig. 3.3. It reveals that the system performance has been significantly improved even with the presence of large non-repetitive disturbance uncertainties. Figure 3.4 shows the control input signals for the four rooms, which indicates that the input saturation limit is always obeyed. Figure 3.5 shows the performance index J in iteration domain compared to ILC with fixed feedback controller. With the same initial feedback controller, both approaches can greatly improve the performance. The proposed approach, however, achieves smaller asymptotical ILC error. This is because the iteratively tuned feedback controller is optimized with predicted disturbance information and such “customized” design can offer a more aggressive feedback controller for disturbance rejection with the control inputs satisfying the constraints. Note that to prevent occupants from suffering unregulated temperature, the first iteration may be tuned offline in the actual building control systems. The time cost of the proposed algorithm for one iteration is around 120s, which is fairly small compared to the sampling period 30 minutes.

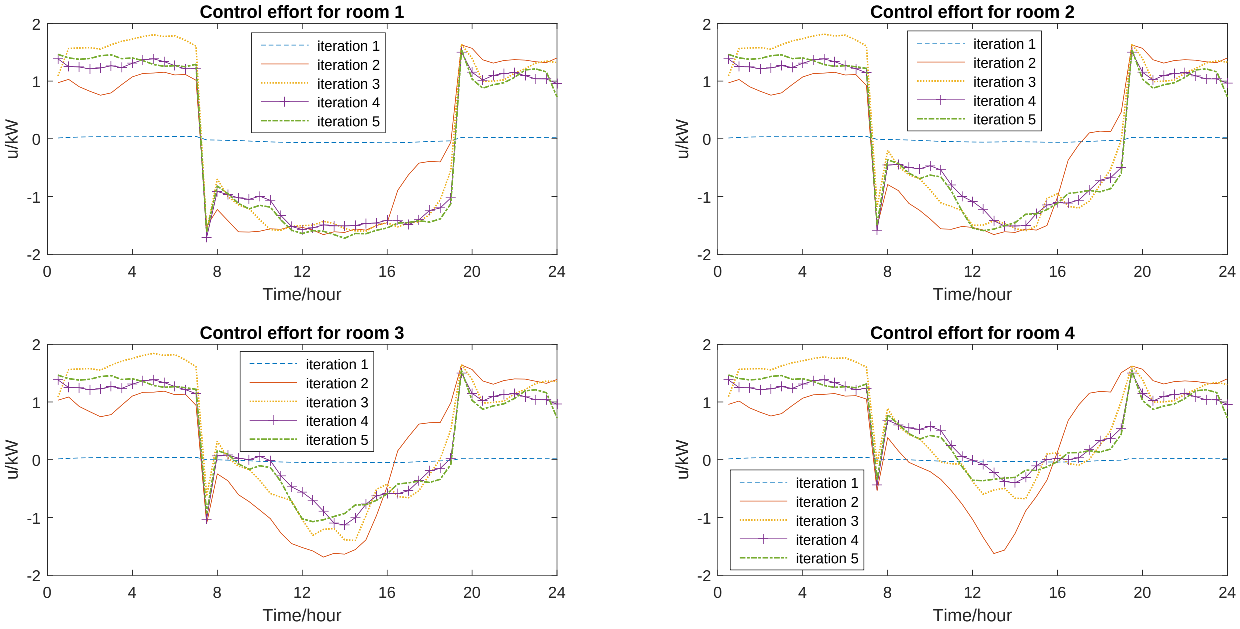

Figure 3.4: Control input of the four rooms in five iterations

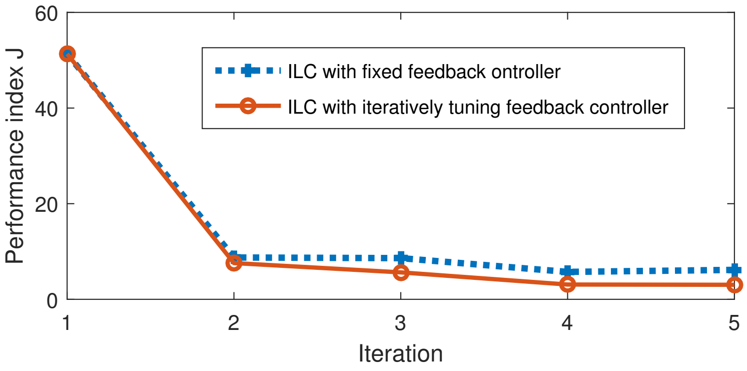

Figure 3.5: Performance comparison with fixed feedback

3.6 Chapter Summary

This chapter presents a centralized control scheme combining constrained current-iteration ILC with iteratively tunable feedback controller. By allowing flexibilities in not only the feedforward ILC signal but also the feedback controllers, both the repetitive disturbance components and the non-repetitive components can be effectively eliminated or attenuated. Moreover, the input saturation limit, ILC convergence and closed-loop stability requirement are considered as constraints. The disturbance uncertainties are also incorporated into the optimization problem for enhanced performance robustness. The simulation results showed that the system performance was greatly improved compared to the standard ILC controller and the time cost was quite acceptable.

Chapter 4

Centralized Concurrent Controller Design

In the previous chapter, a centralized iterative controller design approach is proposed to improve the control performance for multi-zone building thermal systems with heat interaction. However, although the algorithm's performance was verified, the controller results achieved by this method is still an approximation of the optimal solution due to the feedback controller projection step. In order to find the exact optimal feedforward and feedback controller combination, a centralized concurrent controller design is proposed in this chapter, where both controllers solutions are obtained simultaneously through one optimization.

4.1 Introduction

Finding a balance of comfort and energy cost has always been the core task in the building thermal management which has emerged as a popular research topic in recent years. In the building system, a promising candidate for energy saving is the heating, ventilation and air conditioning (HVAC) system. However, the traditional PID controllers used in most existing building systems are not intelligent enough and tuning of parameters is mainly based on the building manager's personal experience. Thus, more advanced and intelligent control techniques for the building HVAC system are in urgent need.

Currently, Model predictive control (MPC) has been the major approach in the building thermal control area due to accessibility to disturbance prediction and building system's relatively long sampling time [25], [1]. In the MPC framework, the controller design is formulated as an optimization problem, and the objective fucntion is optimized over a finite time horizon, but only the current time solution is implemented [15]. In general, an MPC algorithm achieves good performance if the time horizon is reasonably long. This requires that the model and disturbance prediction are accurate over such time horizons. The required model and prediction accuracy, however, may not be available in the real system, especially for a complex plant such as the building system.

It is noted that the weather is repetitive from one day to the next during a relatively long period (one season or one month), especially in tropics such as Singapore. Therefore, in order to reduce the influence of model uncertainties and explicitly take advangtage of the repetitive characteristics of the disturbances, the iterative learning control (ILC) becomes an attractive alternative. The ILC technique has been shown as an effective method to deal with repetitive disturbances [51], [4]. The underlying idea of ILC is to conduct the same task for several times and learn the implicit repeating pattern from experience collected from past iterations. The ILC application in the building thermal control problem has been extensively investigated recently. In [34], a passivity-based ILC is proposed for coupled temperature and humidity control. In [26], the authors combine the ILC with standard MPC to tackle the control task of heating systems.

In this chapter, a concurrent convex optimization-based design of feedforward ILC and feedback controller is proposed. In addition to the feedforward ILC controller, a feedback controller design based on the Youla parametrization is applied to enhance control system's robustness against disturbance uncertainties. With both controllers described in the lifted form, the controller design is formulated into a time-domain optimization with a cost function having thermal comfort and energy consumption terms. With regard to the constraints, the control saturation limit, closed-loop system stability, controller causality and robustness requirement are included. Compared to the algorithm proposed in the previous chapter, solutions to the two controllers in the new proposed algorithm are simultaneously obtained through only one optimization. No extra computational effort is required so that complexity of the problem is largely reduced. Moreover, the robustness preformance against disturbance is also improved.

The remainder of the chapter is organized as follows. In Section 4.2, the controllers are described mathematically and the concurrent controller design problem is formulated as a non-convex optimization problem. In Section 4.3, the formulated non-convex optimization problem is convexified by applying Youla parametrization and introducing new variables. In section 4.4, simulation results are provided to illustrate the proposed algorithm's effectiveness. Section 4.5 concludes the chapter.

4.2 Optimization Based Concurrent Controller Design

In this section, the concurrent feedforward and feedback controller design problem is formulated into a time-domain optimization via rewriting the signals in lifted form.

Concurrent Controller Design

In building temperature control systems, disturbances mainly come from outside environment and inside human-related thermal activities, both of which are nearly repetitive in a long period (one month or one season). For such a system with nearly repetitive disturbances, the iterative learning control approach serves as an effective solution for regulation and disturbance rejection.

However, the control performance of ILC can be significantly diminished if non-repetitive components in disturbance presents. In fact, non-repetitive disturbances always exist in building systems, e.g. unpredictable weather change and unusual human related thermal activities. Such non-repetitive disturbances must be compensated by the feedback controller. There are mainly three requirements for the design of the feedback controller. The first one is effective rejection of predictable non-repetitive disturbance, information of which can be achieved via weather forecast and schedule of indoor human activities. The second one is robustness against unpredictable disturbance uncertainties and the last but most fundamental one is the guarantee of closed-loop system stability.

The concurrent controller design block diagram is shown in Fig. 4.1 with all the signals written in the lifted form. In Fig. 4.1, k is the iteration index, L_{k+1} and $\mathbf{C}_{k+1}$ are feedforward ILC learning gain and feedback controller to be designed for the next iteration, respectively. $\mathbf{R}$ denotes the desired temperature reference, $\mathbf{E}_k$, $\mathbf{T}_k$, $\mathbf{U}_{ff,k}$ and $\mathbf{U}_k$ represent error signal, temperature output signal, feedforward ILC signal and net control input at iteration k. $\hat{\mathbf{D}}_{k+1}$ and $\mathbf{D}_k$ are disturbance prediction for the next iteration and actual disturbance signal respectively.

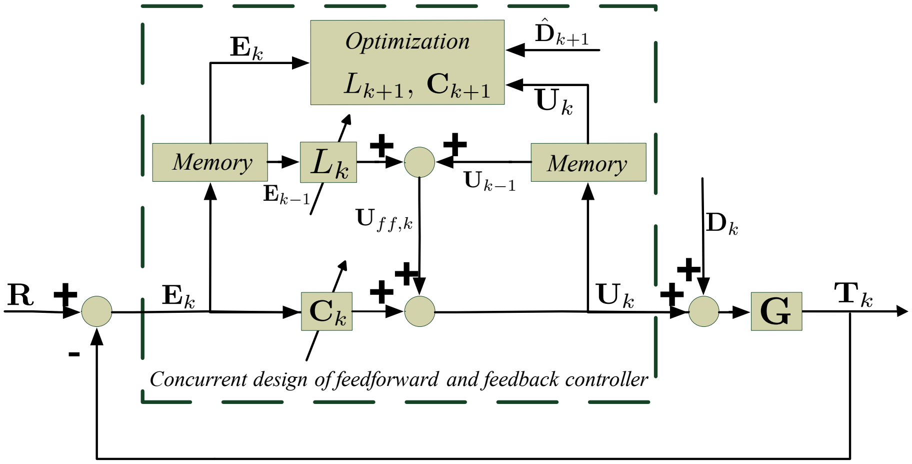

Figure 4.1: Concurrent design of feedforward and feedback controllers

As Fig. 4.1 illustrates, at iteration $k+1$, the feedforward ILC learning gain L_{k+1} and feedback controller $\mathbf{C}_{k+1}$ are updated at the same time according to disturbance prediction $\hat{\mathbf{D}}_{k+1}$, error signal $\mathbf{E}_k$ and control input signal $\mathbf{U}_k$ collected from the last iteration and stored in the system memory. The mathematical expressions for the related signals can be obtained as follows:

$$\mathbf{U}_{ff,k+1} = \mathbf{U}_k + L_{k+1}\mathbf{E}_k, \tag{4.1}$$

$$\mathbf{U}_{k+1} = \mathbf{U}_{ff,k+1} + \mathbf{C}_{k+1}\mathbf{E}_{k+1}, \tag{4.2}$$

$$\mathbf{E}_{k+1} = \mathbf{S}_{k+1}(\mathbf{R} - \mathbf{T}^{in}) - \mathbf{S}_{k+1}\mathbf{G}\mathbf{U}_{ff,k+1} - \mathbf{S}_{k+1}\mathbf{G}\mathbf{D}_{k+1}, \tag{4.3}$$

where

$$\mathbf{S}_{k+1} = (I + \mathbf{G}\mathbf{C}_{k+1})^{-1}, \tag{4.4}$$

corresponds to the system's sensitivity function, I represents the identity matrix with appropriate dimension and $\mathbf{T}^{in}$ is the system's initial state.

Note that the error signal at $k+1$-th iteration $\mathbf{E}_{k+1}$ in (4.3) depends on the future disturbance $\mathbf{D}_{k+1}$ which is not available. Thus, one additional estimated error signal $\hat{\mathbf{E}}_{k+1}$ is introduced as

$$\hat{\mathbf{E}}_{k+1} = \mathbf{S}_{k+1}(\mathbf{R} - \mathbf{T}^{in}) - \mathbf{S}_{k+1}\mathbf{G}\mathbf{U}_{ff,k+1} - \mathbf{S}_{k+1}\mathbf{G}\hat{\mathbf{D}}_{k+1}, \tag{4.5}$$

where $\hat{\mathbf{D}}_{k+1}$ is the disturbance prediction.

Optimization Formulation

In this subsection, the controller design is formulated as a time-domain optimization problem with the feedforwrad ILC learning gain L and feedback controller $\mathbf{C}$ as decision variables.

In order to achieve both satisfactory thermal comfort for the occupants in the building and energy reduction, the objective function J for the optimization is defined as a quadratic one with two weighted terms:

$$J_{k+1} = \hat{\mathbf{E}}_{k+1}^T W_e \hat{\mathbf{E}}_{k+1} + \hat{\mathbf{U}}_{k+1}^T W_u \hat{\mathbf{U}}_{k+1}, \tag{4.6}$$

where k is the iteration index, W_e and W_u are positive definite penalty weights for error signal and control input, respectively. $\hat{\mathbf{U}}_{k+1}$ is the estimated control input defined by

$$\hat{\mathbf{U}}_{k+1} = \mathbf{U}_{ff,k+1} + \mathbf{C}_{k+1}\hat{\mathbf{E}}_{k+1}, \tag{4.7}$$

corresponding to the estimated error signal $\hat{\mathbf{E}}_{k+1}$. In the objective function (4.6), temperature deviation from the desired reference is used to account for thermal comfort level and energy saving is fulfilled by penalizing total energy input to all zones.

With regard to the constraints, first the control saturation constriant is considered. Due to physical limitations of the HVAC system, there exist upper and lower bounds for the energy input to each zone. The upper bound is mainly determined by the maximum opening area of dampers and highest possible supply air temperature of the AHU, i.e., AHU's heating power. Similarly, the lower bound is decided by the AHU's cooling power. Thus, the saturation constraint can be stated as:

$$\underline{U} \leq \hat{\mathbf{U}}_{k+1} \leq \overline{U}, \tag{4.8}$$

where $\underline{U}$ and $\overline{U}$ denote the lower and upper bound of the control input respectively.

Another constraint is imposed to enforce that the temperature responses always lie in the comfort zone defined by lower and upper temperature bounds. The comfort zone constraint can be formulated as

$$\underline{T} \leq \hat{\mathbf{T}}_{k+1} \leq \overline{T}, \tag{4.9}$$

where $\underline{T}$ and $\overline{T}$ are temperature lower and upper bound respectively and

$$\hat{\mathbf{T}}_{k+1} = \mathbf{R} - \mathbf{T}^{in} - \hat{\mathbf{E}}_{k+1}, \tag{4.10}$$

is the estimated temperature output for the $k+1$-th iteration.

Moreover, as the feedback controller is updated for every iteration, the closed-loop system's stability and the feedback controller's causality must be guaranteed which establish another constraint for the optimization. To make the matrix $\mathbf{C}$ able to represent a proper causal feedback controller, $\mathbf{C}$ must be an impulse resposne matrix with the same structure as matrix $\mathbf{H}$ defined in (2.19) with each element as a $z \times z$ matrix, where z is the number of zones in the system. With regard to the closed-loop stability guarantee, the bounded-input-bounded-output (BIBO) stability is applied and in the finite time-horizon case, it can be approximated by

$$||\mathbf{S}_{k+1}\alpha||_1 \leq \gamma_1, \tag{4.11}$$

where α is a column vector of appropriate dimension with the first z elements being one while all other elements being zero and γ_1 is a pre-defined small scalar. By (4.11), the 1-norms of z zones' impulse responses are summed and constrained to be less than a given number γ_1, which guarantees that all zones' responses will not go crazy.

The last constraint considered is the system's robustness against disturbance uncertainties. In the frequency domain, a popular technique to sovle such robust control problems is the H_2 control technique which minimizes the H_2 norm of the transfer fucntion from disturbance to output. In time domain controller design, the H_2 norm idea is still valid. Hence, via imposing certain constraints on the impulse response matrix corresponding to the transfer function from disturbance D to the output T, the control performance can be accordingly adjusted. Mathematically, the robustness constraint can be stated as:

$$||\mathbf{G}_{d2t,k+1}\alpha||_2 \leq \gamma_2, \tag{4.12}$$

where $\mathbf{G}_{d2t,k+1}$ is the impulse resposne matrix corresponding to the trasnfer function from disturbance to output at iteration $k+1$, α is the same vector as defined for the stability constraint (4.11) and γ_2 is another pre-defined scalar reflecting the performance requirement. In fact, $||\mathbf{G}_{d2t,k+1}\alpha||_2$ is the approximation of H_2 norm in the finite time-horizon. $\mathbf{G}_{d2t,k+1}$ can be explicitly written as:

$$\mathbf{G}_{d2t,k+1} = \mathbf{S}_{k+1}\mathbf{G}. \tag{4.13}$$

Combining the defined objective function (4.6) and constraints (4.8) - (4.12), the optimization problem can be formulated as:

$$\begin{aligned} \min_{L_{k+1},\mathbf{C}_{k+1}} \quad & J_{k+1} \\ s.t. \quad & \underline{U} \leq \hat{\mathbf{U}}_{k+1} \leq \overline{U}, \\ & \underline{T} \leq \hat{\mathbf{T}}_{k+1} \leq \overline{T}, \\ & \mathbf{C}_{k+1}\text{is properly structured and causal,} \\ & ||\mathbf{S}_{k+1}\alpha||_1 \leq \gamma_1, \\ & ||\mathbf{G}_{d2t,k+1}\alpha||_2 \leq \gamma_2. \end{aligned} \tag{4.14}$$

However, the optimization (4.14) is non-convex due to coupling of L_{k+1} and $\mathbf{C}_{k+1}$ in error signal term $\hat{\mathbf{E}}_{k+1}$ and control input term $\hat{\mathbf{U}}_{k+1}$. In addition, the feedback controller variable $\mathbf{C}_{k+1}$ also appears in inverse form in the sensitivity matrix term $\mathbf{S}_{k+1}$. Therefore, both the objective function and constraints are non-convex, which makes the problem difficult to solve.

4.3 Convexification based on Youla Parametrization

In this section, the Youla-parametrization is applied to the feedback controller design so that the design task of feedback controller matrix $\mathbf{C}_{k+1}$ is transferred to design of a $\mathbf{Q}$ system which can describe all the stabilizing controllers. Moreover, by introduction of the variable $\mathbf{Q}$ and another new variable, the objective function and constraints of the optimization problem (4.14) are all convexified.

Youla Parametrization

Youla parametrization, also known as Q parametrization first proposed in [57] is a general description of all stabilizing controllers for a given plant system. The general Youla parametrization theorem states that given a MIMO system $P(z^{-1})$ with proper, stable and coprime right factorization $N_p(z^{-1})D_p^{-1}(z^{-1})$ and left factorization $\tilde{D}_p^{-1}(z^{-1})\tilde{N}_p(z^{-1})$, let $X(z^{-1})$, $Y(z^{-1})$ and $\tilde{X}(z^{-1})$, $\tilde{Y}(z^{-1})$ be stable and proper systems that satisfy the Bezout equations, which are:

$$X(z^{-1})D_p(z^{-1}) + Y(z^{-1})N_p(z^{-1}) = I,$$

$$\tilde{N}_p(z^{-1})\tilde{Y}(z^{-1}) + \tilde{D}_p(z^{-1})\tilde{X}(z^{-1}) = I,$$
$$\tilde{D}_p(z^{-1})N(z^{-1}) - \tilde{N}_p(z^{-1})D(z^{-1}) = 0,$$
$$Y(z^{-1})\tilde{X}(z^{-1}) - X(z^{-1})\tilde{Y}(z^{-1}) = 0,$$

then all the stabilizing controllers $K(z^{-1})$ can be parametrized as

$$\begin{aligned} K(z^{-1}) &= \big(X(z^{-1}){-}Q(z^{-1})\tilde{N}_p(z^{-1})\big)^{-1} \\ &\qquad \times\big(Y(z^{-1}){+}Q(z^{-1})\tilde{D}_p(z^{-1})\big) \\ &= \big(\tilde{Y}(z^{-1}){+}D_p(z^{-1})Q(z^{-1})\big)^{-1} \\ &\qquad \times\big(\tilde{X}(z^{-1}){-}N_P(z^{-1})Q(z^{-1})\big), \end{aligned} \tag{4.15}$$

where $Q(z^{-1})$ is an arbitrary stable system.

For the building thermal system discussed in this dissertation, as the plant system is always stable due to its physical characteristics, one intuitive candidate to the systems $X(z^{-1})$, $Y(z^{-1})$, $N_p(z^{-1})$, $D_p(z^{-1})$ and $\tilde{X}(z^{-1})$, $\tilde{Y}(z^{-1})$, $\tilde{N}_p(z^{-1})$, $\tilde{D}_p(z^{-1})$ can be found as

$$X(z^{-1}) = \tilde{X}(z^{-1}) = I, Y(z^{-1}) = \tilde{Y}(z^{-1}) = 0,$$

$$N_p(z^{-1}) = \tilde{N}_p(z^{-1}) = P(z^{-1}), D_p(z^{-1}) = \tilde{D}_p(z^{-1}) = I.$$

Then the general Youla theorem can be simplified and the stabilizing controllers $K(z^{-1})$s' description (4.15) is reduced to

$$\begin{aligned} K(z^{-1}) &= \big(I - Q(z^{-1})P(z^{-1})\big)^{-1}Q(z^{-1}), \\ &= Q(z^{-1})\big(I - P(z^{-1})Q(z^{-1})\big)^{-1}. \end{aligned} \tag{4.16}$$

Applying (4.16) to the building thermal dynamics system (2.16) and rewriting it into the lifted form obtains:

$$\mathbf{C} = (I - \mathbf{QG})^{-1}\mathbf{Q} = \mathbf{Q}(I - \mathbf{GQ})^{-1}, \tag{4.17}$$

where the lifted form signals and systems are in bold form. As shown in (4.17), design of the feedback controller matrix $\mathbf{C}$ is now transferred to design of a stable system $\mathbf{Q}$. This transfer idea is explained in the block diagram form as in Fig. 4.2, where $\mathbf{U}_{fb,k}$ is the feedback control input at iteration k and the state space form of system $\mathbf{C}'$ and $\mathbf{Q}_k$ are given by

$$C' : \begin{cases} \hat{x}(t+1) &= A_g\hat{x}(t) + B_g v_k(t) \\ u_{fb,k}(t) &= v_k(t) \\ y_k(t) &= \hat{x}(t) + e_k(t) \end{cases}, \tag{4.18}$$

$$Q_k : \begin{cases} x_{q,k}(t+1) &= A_{q,k}x_{q,k}(t) + B_{q,k}y_k(t) \\ v_k &= C_{q,k}x_{q,k}(t) + D_{q,k}y_k(t) \end{cases}. \tag{4.19}$$

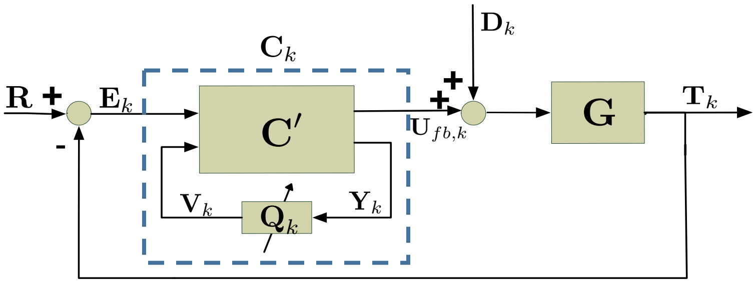

Figure 4.2: Youla parametrization

Convexified Optimization

With the help of Youla parametrization, the feedback controller design has been transferred to design of the Q system. In this subsection, another new variable is introduced and the optimization problem (4.14) is convexified.

First, the following two equations can be derived from (4.17):

$$(I + \mathbf{GC})^{-1} = I - \mathbf{GQ}, \tag{4.20}$$

$$(I + \mathbf{CG})^{-1} = I - \mathbf{QG}. \tag{4.21}$$

Then the error and control input estimation signal $\hat{\mathbf{E}}_{k+1}$ in (4.5) and $\hat{\mathbf{U}}_{k+1}$ in (4.7) can be reformulated as

$$\begin{aligned} \hat{\mathbf{E}}_{k+1} &= \mathbf{R} - \mathbf{T}^{in} + \mathbf{GQ}_{k+1}(\mathbf{GU}_k + \mathbf{G}\hat{\mathbf{D}}_{k+1} - \mathbf{R} + \mathbf{T}^{in}) \\ &\quad -\mathbf{G}(\mathbf{U}_k + \hat{\mathbf{D}}_{k+1}) + (\mathbf{GQ}_{k+1} - I)\mathbf{G}L_{k+1}\mathbf{E}_k, \end{aligned} \tag{4.22}$$

$$\begin{aligned} \hat{\mathbf{U}}_{k+1} &= \mathbf{U}_k + \mathbf{Q}_{k+1}(\mathbf{R} - \mathbf{T}^{in} - \mathbf{G}\hat{\mathbf{D}}_{k+1} - \mathbf{GU}_k) \\ &\quad -\mathbf{G}^{-1}(\mathbf{GQ}_{k+1} - I)\mathbf{G}L_{k+1}\mathbf{E}_k. \end{aligned} \tag{4.23}$$

It can be observed that the only term with $\mathbf{Q}_{k+1}$ and L_{k+1} coupled together in both $\hat{\mathbf{E}}_{k+1}$ and $\hat{\mathbf{U}}_{k+1}$ is $(\mathbf{G}\mathbf{Q}_{k+1}-I)\mathbf{G}L_{k+1}$. Note that although $\mathbf{Q}_{k+1}$ is restricted to be in a proper impulse response matrix form, however, there are no constraints imposed on the structure of L_{k+1}, which results in no specific structural constraints for the matrix $(\mathbf{G}\mathbf{Q}_{k+1}-I)\mathbf{G}L_{k+1}$. Hence, L_{k+1} provides one extra degree-of-freedom (DOF) for design and it is valid to define a new variable as

$$\mathbf{X}_{k+1} = (\mathbf{G}\mathbf{Q}_{k+1}-I)\mathbf{G}L_{k+1}. \tag{4.24}$$

At this stage, $\hat{\mathbf{E}}_{k+1}$ and $\hat{\mathbf{U}}_{k+1}$ can be brought to the convex form with regard to new variables $\mathbf{Q}_{k+1}$ and $\mathbf{X}_{k+1}$ as

$$\begin{aligned}\hat{\mathbf{E}}_{k+1} &= \mathbf{R}-\mathbf{T}^{in}+\mathbf{G}\mathbf{Q}_{k+1}(\mathbf{G}\mathbf{U}_k+\mathbf{G}\hat{\mathbf{D}}_{k+1}-\mathbf{R}+\mathbf{T}^{in})\\ &\quad -\mathbf{G}(\mathbf{U}_k+\hat{\mathbf{D}}_{k+1})+\mathbf{X}_{k+1}\mathbf{E}_k, && (4.25)\\ \hat{\mathbf{U}}_{k+1} &= \mathbf{U}_k+\mathbf{Q}_{k+1}(\mathbf{R}-\mathbf{T}^{in}-\mathbf{G}\hat{\mathbf{D}}_{k+1}-\mathbf{G}\mathbf{U}_k)\\ &\quad -\mathbf{G}^{-1}\mathbf{X}_{k+1}\mathbf{E}_k, && (4.26)\end{aligned}$$

both of which are linear fucntions of $\mathbf{Q}_{k+1}$ and $\mathbf{X}_{k+1}$.

The optimization (4.14) can then be revised as

$$\begin{aligned}\min_{\mathbf{Q}_{k+1},\mathbf{X}_{k+1}} \quad & J_{k+1} && (4.27)\\ s.t. \quad & \underline{U} \le \hat{\mathbf{U}}_{k+1} \le \overline{U},\\ & \underline{T} \le \hat{\mathbf{T}}_{k+1} \le \overline{T},\\ & \mathbf{Q}_{k+1}\text{is proper and causal},\\ & \|\mathbf{Q}_{k+1}\alpha\|_1 \le \gamma_q,\\ & \|\mathbf{G}_{d2t,k+1}\alpha\|_2 \le \gamma_2,\end{aligned}$$

where the new expression of $\mathbf{G}_{d2t,k+1}$ is

$$\mathbf{G}_{d2t,k+1} = \mathbf{G}\mathbf{Q}_{k+1}\mathbf{G}-\mathbf{G},$$

and γ_q is a pre-determined small scalar for Q system stability guarantee. The closed-loop stability constraint (4.11) is replaced by Q stability in (4.27) because a stable Q will always result in a stabilizing controller according to the Youla parametrization theorem. After obtaining the solutions to $\mathbf{Q}_{k+1}$ and $\mathbf{X}_{k+1}$, the ILC learning gain L_{k+1} and the feedback controller $\mathbf{C}_{k+1}$ can be recovered as

$$L_{k+1} = \mathbf{G}^{-1}(\mathbf{G}\mathbf{Q}_{k+1}-I)^{-1}\mathbf{X}_{k+1}, \tag{4.28}$$

and

$$\mathbf{C}_{k+1} = (I-\mathbf{Q}_{k+1}\mathbf{G})^{-1}\mathbf{Q}_{k+1}. \tag{4.29}$$

The proposed convex concurrent feedforward and feedback controller design algorithm can be summarized by the block diagram in Fig. 4.3.

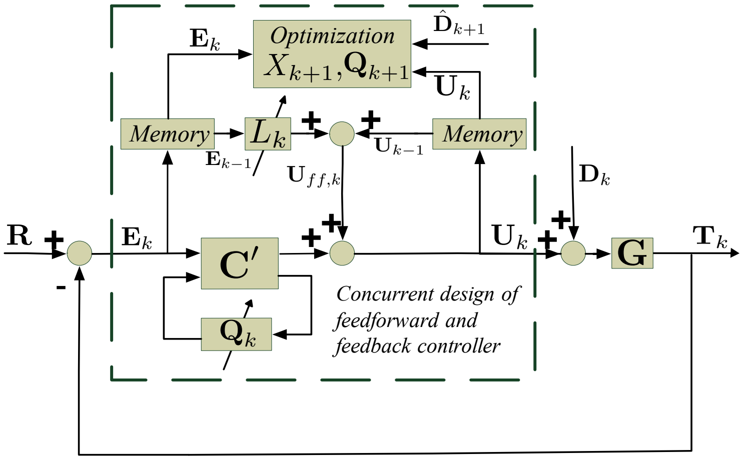

Figure 4.3: Convex concurrent feedforward and feedback controller design

4.4 Simulation Results

In this section, simulation results are provided to demonstrate the proposed concurrent controller design algorithm's effectiveness. In the simulation, the disturbance prediction is assumed to be accessible, but its accuracy is not required.

Simulation Setup

We simulated the proposed algorithm on a testbed system with four rooms as shown in Fig. 2.7 with thermal parameters in Table 2.1. The centralized control structure is selected, which means that the control input to all the rooms are solved at the same time by one optimization. In the simulation, one iteration for the controller design is selected to be one day and the sampling time is chosen as 30 minutes.

Since the application is intended to take place in Singapore, the Singapore weather forecast data [14] and a self-defined human activity schedule are used for the disturbance prediction while the real weather data and perturbed human related thermal signal are treated as real disturbance. Figure 4.4 shows the net effect of the disturbance sources for one day in the form of temperature and makes a comparison between the prediction and real disturbance data. It is shown in Fig. 4.4 that there exists obvious difference between the prediction and real data, which verifies the necessity of a robust feedback controller.

With regard to the reference, within the working hours from 7 : 30am to 6pm, each room's desired temperature was set as shown in Table 4.1 and no reference was defined for the rest time for energy saving purpose.

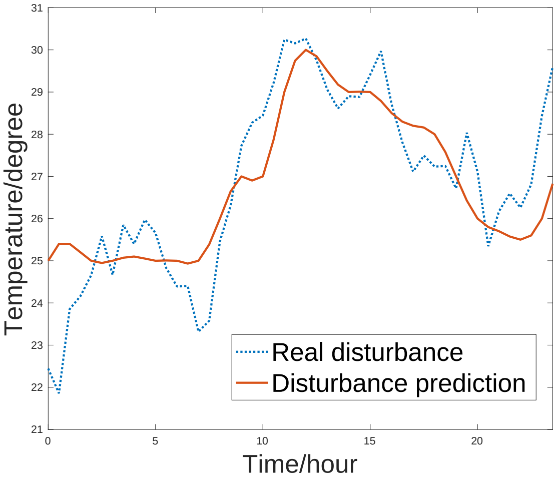

Figure 4.4: Comparison between disturbance prediction and real disturbance

Table 4.1: Desired temperature during working hours

Room	1	2	3	4
Temperature(°C)	22.5	22	23	23.5

Discussion of Simulation Results

First, the proposed algorithm was simulated assuming that the disturbance prediction was perfectly accurate and compared to the ILC with a fixed PID feedback controller. The performance of the concurrently designed controllers is shown in Fig. 4.5, where the temperature output profiles for room 1 and room 2 in three consecutive days are depicted. Note that the disturbances for these three days are not the same, which means non-repetitiveness still exists in the system. In the first iteration, as there is no pre-knowledge available, the performance was very poor. However, after the first iteration, the performance was significantly improved as both feedforward ILC and feedback controller have come into effect so that both repetitive and non-repetitive disturbance components were properly attenuated. Besides, the precooling behavior instead of sharp cooling was also observed after the first iteration since ILC has learned reference information and thus automatically selected the best energy saving strategy. Figure 4.6 shows the performance of pure ILC with fixed pre-designed PID feedback controller. Comparison of Fig. 4.5 and Fig. 4.6 explicitly illustrates that the introduced concurrently updated feedback controller greatly reduced the influence of non-repetitive disturbances.

Then, the disturbance uncertainties were added to the system so that the disturbance prediction was no more accurate as shown in Fig. 4.4. The corresponding simulation results of the proposed algorithm are shown in Fig. 4.7, where the desired temperatures were well tracked during the working hours and the whole day temperature profiles were within the bounded comfort zone. As

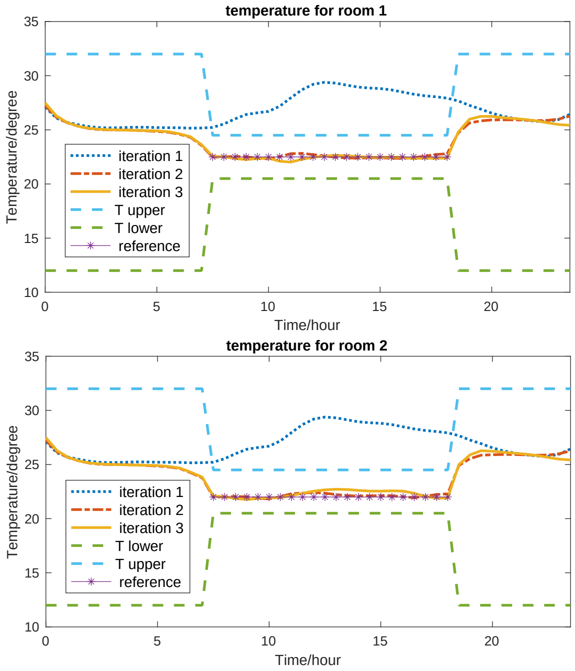

Figure 4.5: Concurrent controller design with accurate disturbance prediction

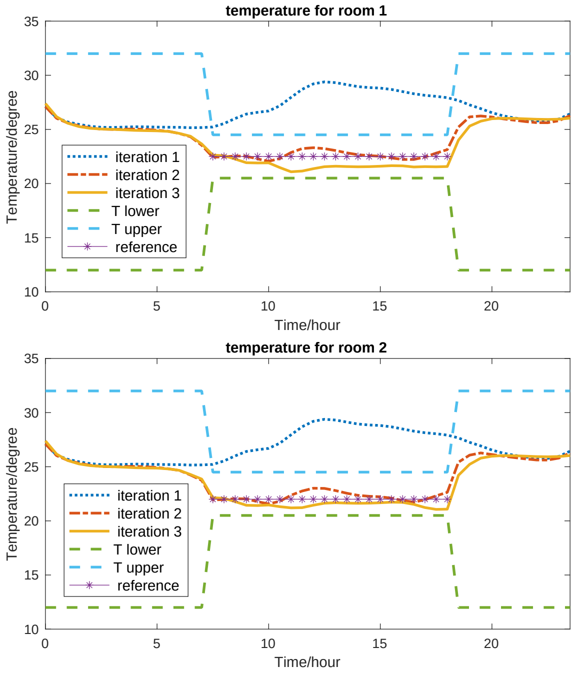

Figure 4.6: ILC with fixed feedback controller given accurate disturbance prediction

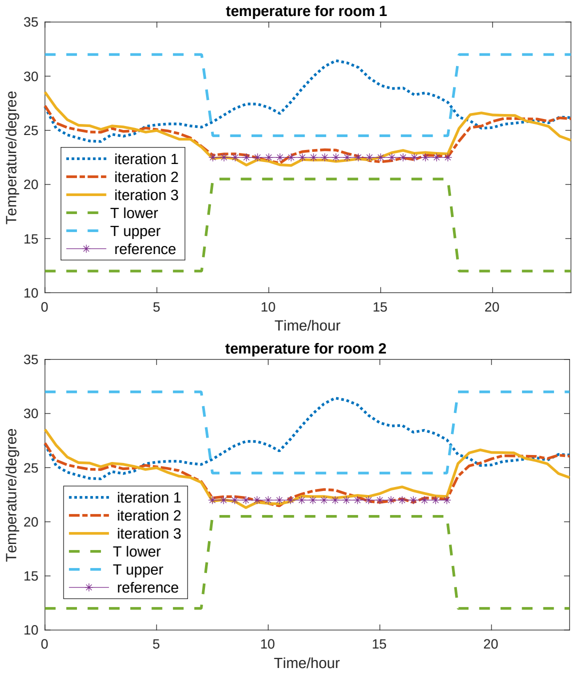

Figure 4.7: Concurrent controller design with disturbance uncertainties

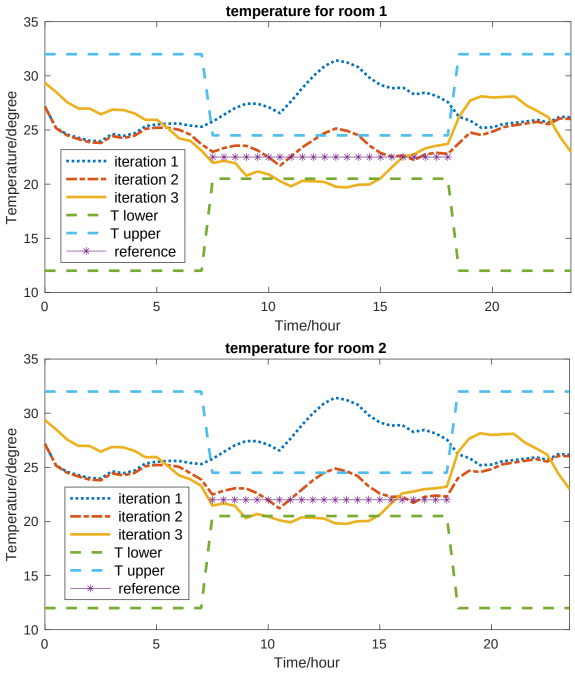

Figure 4.8: ILC with fixed feedback controller and disturbance uncertainties

a comparison, Fig. 4.8 shows the result of pure ILC design with fixed feedback controller, where the reference tracking performance was not satisfactory and the thermal comfort constraints were also violated.

4.5 Chapter Summary

In this chapter, a centralized concurrent feedforward and feedback controller design algorithm was proposed. In order to reject repetitive disturbance components, the iterative learning control was utilized for the feedforward controller. With regard to the non-repetitive components, a Youla-parametrized feedback controller was designed. The controller design was formulated into a convex time-domain optimization that took both thermal comfort and energy consumption into account. The optimal feedforward ILC learning gain and feedback controller combination was solved simultaneously by the optimization without assumptions on the feedback controller structure made in the iterative design proposed in Chapter 3. Benefiting from the general feedback structure described in Youla form, the concurrent algorithm would potentially achieve better control performance compared to the iterative method. To make the algorithm implementable for the practical system, the control saturation limit, thermal comfort demand, closed-loop stability guarantee and robustness against disturbance uncertainties were included as constraints.

Chapter 5

Distributed Cooperative Controller Design

Chapter 3 and 4 focus on the optimal centralized control for the building thermal system. However, the actual building system may contain hundreds of room, for such large-scale system, the computational power may not be strong enough to find an optimal centralized controller solution. Actually, the large-scale centralized system can be separated into several distributed and cooperative small-scale subsystems that communicate among each other. In this chapter, a distributed and cooperative optimization-based iterative learning control (ILC) algorithm is proposed for the large-scale building temperature control problem to handle interactions between zones and reject disturbances. With the algorithm, the large-scale building temperature control problem becomes solvable with reasonable computational load and guaranteed control performance under nearly repetitive disturbances. For each subsystem, a convex optimization problem is solved. The cooperative learning policy allows all subsystems to contribute together to improve the overall performance.

5.1 Introduction

Recently, research about building control systems, especially the heating, ventilation and air conditioning (HVAC) system has received a great amount of attention. In order to utilize energy more efficiently, several methodologies have been proposed to replace the traditional simple structured feedback controllers for HVAC systems. One popular approach is model predictive control (MPC) [25], [1], which aims to optimize the energy consumption with constraints. To achieve good control performance, a long time horizon is generally necessary for MPC, which leads to high reliance on the model accuracy of both thermal dynamics and disturbance prediction. However, in practice, high model accuracy cannot be always guaranteed.

An alternative to MPC is the iterative learning control (ILC). The essential idea of ILC is to improve system's performance by learning its repetitive characteristics through past iterations experience [51], [4]. For the HVAC control system, both the reference and disturbances, including ambient weather and inside human activities, exhibit certain repetitive properties. These repetitive features endorse the application of ILC to the building control problem. In [33], a frequency domain ILC is designed for the building temperature control problem. In our previous work [42] and [43], a two-degree-of-freedom controller is proposed to simultaneously handle both repetitive and non-repetitive disturbance using combined ILC and iterative feedback tuning (IFT) technique. However, these methods are all centralized and thus quite time-consuming or even intractable for large-scale systems where hundreds of zones need to be controlled.

To deal with the scalability challenge, some distributed MPC algorithms have been proposed, e.g., [56], [36]. In the distributed MPC framework, the centralized system is partitioned into several subsystems that communicate. Each sub-system MPC solves a small-scale optimization problem. By integrating all subsystems' solutions, the interactions among the subsystems can be handled. However, this approach leads to higher dependency on prediction accuracy than normal MPC. In the ILC framework, similar subsystem concepts are utilized in cooperative ILC for formation control problems [31], [20]. In cooperative ILC, each subsystem can learn from not only its own past experience but also its neighbouring control zones'. In this way, the control performance can be improved and more information can be leveraged. However, it ignores possible interactions among subsystems, which makes it impractical to be directly applied to building control problems where the interactions are strong.

In this chapter, a distributed cooperative optimization-based ILC algorithm is proposed to address the large-scale HVAC system control problem. The large-scale centralized system is separated into several subsystems, where in each subsystem, a small-scale convex optimization is solved to get the optimal learning gains. Compared to original cooperative ILC, the significant difference, however, lies in the incorporation of the interaction effects into the overall performance evaluation and each of the small-scale optimization. With this algorithm, physical constraints and ILC convergence constraints have been considered, and the control performance depends less on the model accuracy but more on the past iteration data. To authors' best knowledge, the distributed cooperative ILC approach has not been applied to the HVAC control problem before.

The chapter is organized as follows. In Section 5.2, the cooperative ILC is introduced and its convergence properties are analyzed. In Section 5.3, the building temperature control problem is formulated as a convex optimization problem. In Section 5.4, the detailed distributed cooperative ILC algorithm is described and convergence analysis of the whole algorithm is conducted. In section 5.5, several simulation results are presented to illustrate the proposed algorithm's effectiveness. Section 5.6 concludes the chapter.

5.2 Cooperative ILC

In this section, the cooperative ILC is applied to the system and the convergence analysis is conducted using the lifted system technique described in 2.5.

Cooperative ILC Algorithm

In building temperature control systems, disturbances mainly come from ambient heat and inside heat generated by human activities, both of which are nearly repetitive in a long period (one month or one season). For such a system with nearly repetitive disturbances, the iterative learning control approach serves as a quite effective solution for regulation and disturbance rejection. The central idea of ILC is to learn from past experience and improve performance of the next iteration.

Moreover, in order to take the most advantage of available information and solve the problem, we propose to use cooperative ILC instead of traditional I-type learning ILC so that each zone can learn from not only its own experience but also the neighbours'. The main difference of cooperative ILC and original ILC lies in the definition of error signal. By taking relative temperature information into account, the new error signal for zone i is defined as:

$$\begin{aligned} e_{i,k}(t) &= T_i^{ref}(t) - T_{i,k}(t) + \sum_{j\in\mathbf{N}_i}\Big[(T_i^{ref}(t)- \\ & \qquad T_{i,k}(t)) - (T_j^{ref}(t) - T_{j,k}(t))\Big], \end{aligned} \tag{5.1}$$

where $e_{i,k}$ denotes the error signal of zone i at iteration k, T_i^{ref} denotes the reference temperature profile of zone i and $\mathbf{N}_i$ represents the neighboring zone set of i.

With the error signal defined, the cooperative ILC update law can be expressed by

$$u_{i,k+1}^{ILC}(t) = u_{i,k}(t) + \Psi_{i,k}(q)^{(t)} e_{i,k}(t), \tag{5.2}$$

where $u_{i,k+1}^{ILC}(t)$ denotes feedforward ILC input to zone i at iteration $k+1$, q^{-1} is the backward shift operator and

$$\Psi_{i,k}(q)^{(t)} = \psi_{i,k}^0(t) + \psi_{i,k}^1(t)q^{-1} + \cdots + \psi_{i,k}^{N-1}(t)q^{-(N-1)}$$

is zone i's learning function at iteration k. Note that the t index is not omitted as the learning gains are not necessary to be constant during one iteration, which provides more degree of freedom for controller design.

Using the lifting technique, the ILC input in (5.2) can be represented by

$$\underbrace{\begin{bmatrix} u_{i,k+1}^{ILC}(0) \\ u_{i,k+1}^{ILC}(1) \\ \vdots \\ u_{i,k+1}^{ILC}(N-1) \end{bmatrix}}_{\mathbf{u}_{i,k+1}^{ILC}} = \underbrace{\begin{bmatrix} u_{i,k}(0) \\ u_{i,k}(1) \\ \vdots \\ u_{i,k}(N-1) \end{bmatrix}}_{\mathbf{u}_{i,k}} + L_{i,k} \underbrace{\begin{bmatrix} e_{i,k}(1) \\ e_{i,k}(2) \\ \vdots \\ e_{i,k}(N) \end{bmatrix}}_{\mathbf{e}_{i,k}}, \tag{5.3}$$

with the learning matrix defined as:

$$L_{i,k} = \begin{bmatrix} \psi_{i,k}^0(0) & 0 & \cdots & 0 \\ \psi_{i,k}^1(1) & \psi_{i,k}^0(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \psi_{i,k}^{N-1}(N-1) & \psi_{i,k}^{N-2}(N-1) & \cdots & \psi_{i,k}^0(N-1) \end{bmatrix}.$$

Convergence Analysis of Cooperative ILC

In this subsection, the convergence condition of the cooperative ILC is developed. First, some notations are introduced. The adjacency matrix $\Omega = \Omega_{ij}$ is defined as $\Omega_{ij} = 1$ if zone i and zone j are adjacent or $i = j$ and $\Omega_{ij} = 0$ otherwise. Φ is a matrix converting stacked subsystem vector $[\mathbf{T}_{1,k}^T, \cdots, \mathbf{T}_{n,k}^T]^T$=$[T_{1,k}(0), \cdots, T_{1,k}(N), \cdots, T_{n,k}(0), \cdots, T_{n,k}(N)]^T$ to centralized system lifted vector $\mathbf{T}_k$=$[T_{1,k}(0), \cdots, T_{n,k}(0), \cdots, T_{1,k}(N), \cdots, T_{n,k}(N)]^T$, defined as $\Phi = \Phi_{ij}$, $\Phi_{ij} = 1$ if $j = (mod(i,N) - 1) * n + \lceil i/N \rceil$ ($\lceil \bullet \rceil$ denotes the rounddown function) or $j = (N-1) * n + i/N$ and $mod(i,N) = 0$, $\Phi_{ij} = 0$ otherwise.

With these notations defined, the lifted system representation of the centralized cooperative ILC can be obtained:

$$\begin{aligned} \mathbf{u}_{k+1}^{ILC} &= \mathbf{u}_k + \Phi \mathbf{L}_k \mathbf{e}_k, \\ &= \mathbf{u}_k + \Phi \mathbf{L}_k (\Omega \otimes I_N) \Phi \mathbf{E}_k \end{aligned} \tag{5.4}$$

where

$$\mathbf{L}_k = \begin{bmatrix} L_{1,k} & 0 & \cdots & 0 \\ 0 & L_{2,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & L_{n,k} \end{bmatrix},$$

$\mathbf{u}_{k+1}^{ILC} =$

$$[u_{1,k+1}^{ILC}(0), \cdots, u_{n,k+1}^{ILC}(0), \cdots, u_{1,k+1}^{ILC}(N), \cdots, u_{n,k+1}^{ILC}(N)]^T,$$

$\mathbf{e}_k = [\mathbf{e}_{1,k}^T, \cdots, \mathbf{e}_{n,k}^T]^T$ and $\mathbf{E}_k = \mathbf{T}^{ref} - \mathbf{T}_k$. Operator $\otimes$ denotes the Kronecker product and I_N is $N \times N$ identity matrix. The matrix Φ's function is to convert the stacked subsystem lifted vector with the structure of $[T_{1,k}(0), \cdots, T_{1,k}(N), \cdots, T_{n,k}(0), \cdots, T_{n,k}(N)]^T$ to centralized system lifted vector in the form of $[T_{1,k}(0), \cdots, T_{n,k}(0), \cdots, T_{1,k}(N), \cdots, T_{n,k}(N)]^T$.

With (5.4), the closed-loop error signal at iteration $k+1$ can be reformulated as:

$$\begin{aligned} \mathbf{E}_{k+1} &= (I + \mathbf{GC})^{-1}(I - \mathbf{G}\Phi\mathbf{L}_k(\Omega \otimes I_N)\Phi)\mathbf{E}_k \\ &\quad + (I + \mathbf{GC})^{-1}\mathbf{G}\delta, \end{aligned} \tag{5.5}$$

where $\mathbf{C}$ denotes the feedback controller's transfer function matrix and disturbance difference between iterations is merged into δ term to facilitate analysis.

The ILC convergence requires $\|\mathbf{E}_{k+1}\|_2 \leq \|\mathbf{E}_k\|_2$ when disturbance is purely repetitive, i.e. $\delta = 0$. From (5.5), it can be seen that this condition is equivalent to

$$\rho((I + \mathbf{GC})^{-1}(I - \mathbf{G}\Phi\mathbf{L}_k(\Omega \otimes I_N)\Phi)) \leq 1, \tag{5.6}$$

where ρ represents the spectral radius of the system, which is also equal to the system's maximum singular value. As $(I + \mathbf{GC})^{-1}$ and $\mathbf{L}_k$ are both lower triangular matrices, their singular values are the diagonal entries. Therefore, in a more concise manner, (5.6) can be reformulated as:

$$\max_{j=1,2,\cdots,n \times N} \rho_0 |1 - \lambda_{1,j} \mathbf{L}_k^j \lambda_{2,j}| \leq 1, \tag{5.7}$$

where ρ_0 is the diagonal entry of matrix $(I+\mathbf{GC})^{-1}$, $\lambda_{1,j}$'s and $\lambda_{2,j}$'s are singular values of matrix $\mathbf{G\Phi}$ and $(\Omega\otimes I_N)\Phi$ respectively and $\mathbf{L}_k^j$ is the j-th diagonal element of $\mathbf{L}_k$.

Note that both $\mathbf{G\Phi}$ and $(\Omega\otimes I_N)\Phi$ are known constant matrices, hence all the $\lambda_{1,j}$'s and $\lambda_{2,j}$'s can be obtained. Thus, (5.7) actually formulates a convex constraint on $\mathbf{L}_k$, which can be easily implemented in the optimization problem.

5.3 Optimization Problem Formulation

In this section, the cooperative ILC design is formulated as a convex constrained optimization problem in ILC iteration domain.

Centralized Optimization

First let us consider the centralized control structure. It provides a globally optimal solution to the problem. Similar with $\mathbf{E}_{k+1}$, the control input signal $\mathbf{u}_{k+1}$ can be written as:

$$\mathbf{u}_{k+1} \quad = \quad (I+\mathbf{CG})^{-1}[\mathbf{u}_{k+1}^{ILC}+\mathbf{C}(\mathbf{T}^{ref}-\mathbf{T}^{in}-\mathbf{G}\mathbf{w}_{k+1})]. \tag{5.8}$$

The objective function J_{k+1} can then be defined as:

$$J_{k+1}(\mathbf{L}_k) = \mathbf{E}_{k+1}^T W_e \mathbf{E}_{k+1} + \mathbf{u}_{k+1}^T W_u \mathbf{u}_{k+1}, \tag{5.9}$$

where W_e and W_u are positive definite penalty matrices for the error signal and control input signal, respectively. Note that here we assume the feedback controller to be fixed in ILC iteration domain.

With regard to constraints, the most two important constraints for the building temperature control problem are the saturation constraint imposed on control input and the comfort constraint on each zone's temperature profile, which respectively can be formulated as:

$$\underline{U} \leq \mathbf{u}_{k+1} \leq \overline{U}, \tag{5.10}$$

and

$$\underline{T} \leq \mathbf{T}_{k+1} \leq \overline{T}, \tag{5.11}$$

where $\underline{U}$, $\overline{U}$ and $\underline{T}$, $\overline{T}$ are correspondingly lower and upper bounds for control input $\mathbf{u}$ and temperature $\mathbf{T}$.

Summarizing (5.9)-(5.11) and ILC convergence constraint (5.7), the centralized optimization problem can be stated as:

$$\begin{aligned} \min_{\mathbf{L}_k} \quad & J_{k+1} \\ s.t. \quad & \underline{U} \leq \mathbf{u}_{k+1} \leq \overline{U} \\ & \underline{T} \leq \mathbf{T}_{k+1} \leq \overline{T} \\ & \max_{j=1,2,\cdots,n\times N} \rho_0 |1-\lambda_{1,j}\mathbf{L}_k^j \lambda_{2,j}| \leq 1. \end{aligned} \tag{5.12}$$

The optimization problem (5.12) is convex with regard to the learning matrices $L_{i,k}$, for $i = 1,2,\cdots,n$. However, its computational demand grows exponentially with the system size n. In practice, solving and implementing such a centralized optimization-based controller is quite time-consuming and requires more powerful hardware. Such disadvantages greatly restrict the application of centralized controller structure. In addition, the centralized controller also leads to inflexibility, meaning that any damage of the controller will jeopardize the whole system.

Distributed Optimization

In order to achieve similar control performance with centralized controller and acceptable computational efficiency, the distributed cooperative controller is proposed. In the distributed cooperative control framework, each subsystem solves a small-scale optimization problem. The cost function is the same as centralized case and communication between subsystems is allowed. The distributed cooperative optimization problem is defined as:

$$\begin{aligned} \min\nolimits_{L_{i,k}} \quad & J_{k+1} \\ s.t. \quad & \underline{U} \leq \mathbf{u}_{k+1} \leq \overline{U} \\ & \underline{T} \leq \mathbf{T}_{k+1} \leq \overline{T} \\ & \max_{j=1,2,\cdots,N} \rho_0 |1 - \lambda_{1,j} L_{i,k}^{j} \lambda_{2,j}| \leq 1, \end{aligned} \tag{5.13}$$

where $L_{i,k}^{j}$ denotes j-th diagonal entry of $L_{i,k}$, $\lambda_{1,j}$, $\lambda_{2,j}$ represent the corresponding singular values of $\mathbf{G}\Phi$ and $(\Omega \otimes I_N)\Phi$ with regard to $L_{i,k}^{j}$.

Note that as

$$\begin{aligned} \mathbf{u}_{k+1} &= [u_{1,k+1}(0),\cdots,u_{i,k+1}(0),\cdots,u_{n,k+1}(0), \\ &\quad \cdots,u_{1,k+1}(N-1),\cdots,u_{n,k+1}(N-1)]^T, \end{aligned}$$

$\mathbf{E}_{k+1}$ and $\mathbf{T}_{k+1}$ also share the similar structure, information about the other subsystems is involved in both the cost function and the constraints. However, compared to centralized structure, the decision variables in (5.13) are reduced from $\mathbf{L}_k$ for all n subsystems to $L_{i,k}$ for only one subsystem.

5.4 Distributed Cooperative ILC Algorithm and Convergence Analysis

Algorithm

With communication enabled, (5.13) can be solved iteratively. The detailed algorithm of distributed cooperative ILC is described in Algorithm 1, where k denotes ILC iteration and k^c denotes distributed optimization iteration. In the algorithm, each subsystem solves a small-scale optimization-based cooperative ILC design problem after receiving information of other subsystems and then broadcasts its solution to the whole system. By iteratively exchanging information, a near-optimal solution can be found.

Algorithm 1 Distributed Cooperative ILC

```
Given $u_{i,1}^{ILC} \leftarrow 0$, $\omega_i > 0$, $\sum_i \omega_i = 1$, $i = 1,2,\cdots,n$
$k_{max}^c > 0$, $0 < \varepsilon \ll 1$, $\Gamma \gg \varepsilon$
**for** each $k = 1,2,\cdots,m$ **do**
    Given $\mathbf{u}_{i,k+1}^{ILC(k^c)} \leftarrow \mathbf{u}_{i,k}^{ILC}$, $\gamma_i \leftarrow \Gamma$, $i = 1,2,\cdots,n$, $k^c \leftarrow 1$
    $\forall i = 1,2,\cdots,n$, calculate $\mathbf{u}_{i,k}$ according to $\mathbf{u}_{j,k}^{ILC}$'s
    $(j = 1,2,\cdots,n)$
    Implement $\mathbf{u}_{i,k}$'s to the subsystems and record
    temperature information $\mathbf{T}_{i,k}$'s; Calculate $\mathbf{e}_{i,k}$'s
    **while** $\gamma_i > \varepsilon$ and $k^c < k_{max}^c$ **do**
        **for** each $i = 1,2,\cdots,n$ **do**
            $L_{i,k}^{k^c}$ = arg(optimization(5.13))
            $\mathbf{u}_{i,k+1}^{ILC(k^c,*)} \leftarrow \mathbf{u}_{i,k} + L_{i,k}^{k^c}\mathbf{e}_{i,k}$
        **end for**
        **for** each $i = 1,2,\cdots,n$ **do**
            $\mathbf{u}_{i,k+1}^{ILC(k^c+1)} \leftarrow \omega_i \mathbf{u}_{i,k+1}^{ILC(*)} + (1-\omega_i)\mathbf{u}_{i,k+1}^{ILC(k^c)}$
            $\gamma_i \leftarrow \|\mathbf{u}_{i,k+1}^{ILC(k^c+1)} - \mathbf{u}_{i,k+1}^{ILC(k^c)}\|_2$
            Transmit $\mathbf{u}_{i,k+1}^{ILC(k^c+1)}$ to all the other subsystems
            $j = 1,2,\cdots,n, j \neq i$
        **end for**
        $k^c$=$k^c$+1
    **end while**
    **for** each $i = 1,2,\cdots,n$ **do**
        $u_{i,k+1}^{ILC}$=$u_{i,k+1}^{ILC(k^c)}$
    **end for**
**end for**
```

Remark 1: Note that when solving the optimization problem (5.13), the future information of ambient temperature and inside disturbance is required. These unaccessible information is replaced by weather forecast and human heat disturbance prediction in implementation.

Remark 2: The parameter k_{max}^c is used to force termination of the distributed optimization to guarantee algorithm efficiency when dealing with systems that have a great number of subsystems. Although in this case, convergence to the centralized solution cannot be guaranteed, the algorithm can still provide acceptable performance. Details of proof will be shown in the next subsection.

Convergence Analysis

In this subsection, the convergence properties of distributed cooperative ILC will be analyzed. The Algorithm 1 can be separated into two parts, which are cooperative ILC and distributed optimization. As the convergence of cooperative ILC has been proved in the previous section, we will focus on the distributed optimization part here. The following theorem similar to [56] can be established:

Theorem I: With Algorithm 1, the cost function J_{k+1} is non-increasing with distributed optimization iteration k^c.

Proof: By definition of ω_i and $\mathbf{u}_{i,k+1}^{ILC(k^c+1)}=\omega_i\mathbf{u}_{i,k+1}^{ILC(*)}+(1-\omega_i)\mathbf{u}_{i,k+1}^{ILC(k^c)}$ (see Algorithm 1), the following equation can be obtained:

$$\begin{aligned}
& J_{k+1}(\mathbf{u}_{1,k+1}^{ILC(k^c+1)},\cdots,\mathbf{u}_{i,k+1}^{ILC(k^c+1)},\cdots,\mathbf{u}_{n,k+1}^{ILC(k^c+1)}) \\
= \;& J_{k+1}(\omega_1\mathbf{u}_{1,k+1}^{ILC(k^c,*)}+(1-\omega_1)\mathbf{u}_{1,k+1}^{ILC(k^c)},\cdots, \\
& \quad \omega_n\mathbf{u}_{n,k+1}^{ILC(k^c,*)}+(1-\omega_n)\mathbf{u}_{n,k+1}^{ILC(k^c)}) \\
= \;& J_{k+1}(\omega_1\mathbf{u}_{1,k+1}^{ILC(k^c,*)}+\omega_2\mathbf{u}_{1,k+1}^{ILC(k^c)}+\cdots+\omega_n\mathbf{u}_{1,k+1}^{ILC(k^c)},\cdots, \\
& \quad \omega_n\mathbf{u}_{n,k+1}^{ILC(k^c,*)}+\omega_1\mathbf{u}_{n,k+1}^{ILC(k^c)}+\cdots+\omega_{n-1}\mathbf{u}_{n,k+1}^{ILC(k^c)})
\end{aligned}$$

As J_{k+1} is convex, the following inequality can be developed:

$$\begin{aligned}
\leq \;& \sum_i^n \omega_i J_{k+1}(\mathbf{u}_{1,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{i,k+1}^{ILC(k^c,*)},\mathbf{u}_{i+1,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{n,k+1}^{ILC(k^c)}) \\
\leq \;& \sum_i^n \omega_i J_{k+1}(\mathbf{u}_{1,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{i,k+1}^{ILC(k^c)},\mathbf{u}_{i+1,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{n,k+1}^{ILC(k^c)}) \\
= \;& J_{k+1}(\mathbf{u}_{1,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{i,k+1}^{ILC(k^c)},\cdots,\mathbf{u}_{n,k+1}^{ILC(k^c)})
\end{aligned}$$

In addition, as the cost function is in quadratic form, its value is bound to be greater or equal to zero. Thus, theorem I assures the convergence of cost function in the distributed optimization iteration domain. Note that due to penalty on control input, the cost function will not converge to zero.

5.5 Simulation Results

To verify the effectiveness of the proposed algorithm, simulation results are shown in this section. The simulation was run on the test-bed system with four connected zones shown in Fig. 2.7 with parameters in Table 2.1. The reference temperatures are only defined for working hours from 8am to 7pm. Each zone's reference temperature is the same as listed in Table 4.1. For other periods, no references are set.

The weather disturbance and forecast data used for simulation was collected from EnergyPlus energy simulation software website [14] and the inside human disturbance was assumed to be constant. Note that in practice, room schedules are accessible, which will indicate future human activities and this information can be utilized to predict the heat disturbance term using human heat generation model. During the simulation, weather forecast was utilized when solving for ILC input of the next future iteration and real disturbance was introduced into the system along with ILC controller of the current iteration.

In simulation, one ILC iteration is set to be one day (24 hours) and the feedforward ILC controller is updated at the end of every iteration. To comply with definition of references, the penalty matrix for error signal W_e is defined to be a diagonal matrix. Its i-th diagonal entry is equal to u_{max} (the upper bound of control input) if the corresponding time instant of i is in the working period. All the other elements of W_e are set to zeros. u_{max} is selected to balance between energy consumption and comfort demand. W_u is defined as identity matrix of size $nN \times nN$. The learning weights ω_i's are set to $1/n$ for all zones.

All simulations were performed using Matlab 2016b on a laptop lenovo yoga 900 (2.5 GHz Intel Corei7). In the simulation, the algorithm was run in series with regard to the subsystems. The average time cost for one ILC iteration was 8.0314 seconds. In real implementation, as each subsystem could run the algorithm in parallel, the time cost of one ILC iteration is supposed to be around 2 seconds. As a comparison, the centralized optimization was run on the same platform, average time cost of which was 10.2076 seconds for one ILC iteration. It can be seen that the proposed algorithm has achieved much better computational efficiency than the centralized approach for this test case with only $n = 4$ zones. For a practical building system with hundreds of rooms, more significant improvement can be predicted.

Figure 5.1 shows the temperature output of the four control zones in three ILC iterations, each of which tracks the reference temperature quite well during the working periods after the first ILC iteration. In addition, the figures also show that precooling behavior was learned as temperature began to decrease even before the working period started (8am).

Figure 5.2 shows the corresponding control commands for the four rooms. As can be seen from the figures, all the control signals obeyed the saturation constraints and in order to save energy, during the off-work hours, control signals almost stayed at zero. Moreover, the precooling behavior can be seen more clearly in the control signal figures, where control signals came into effect several time steps before the start of references (8am).

Figure 5.3 shows the algorithm's performance w.r.t. distributed optimization iteration k^c when ILC iteration $k = 1$. It can be seen that the cost function kept decreasing until the convergence criteria was met.

5.6 Chapter Summary

In this chapter, a distributed cooperative ILC algorithm was proposed. Compared to the centralized algorithms in Chapter 3 and Chapter 4, the distributed approach significantly reduced the computational cost for the large-scale building control problem. The feedforward ILC controller was designed using lifting technique and solved by a convex optimization problem with thermal comfort constraint, control saturation constraint and ILC convergence constraint. With the proposed

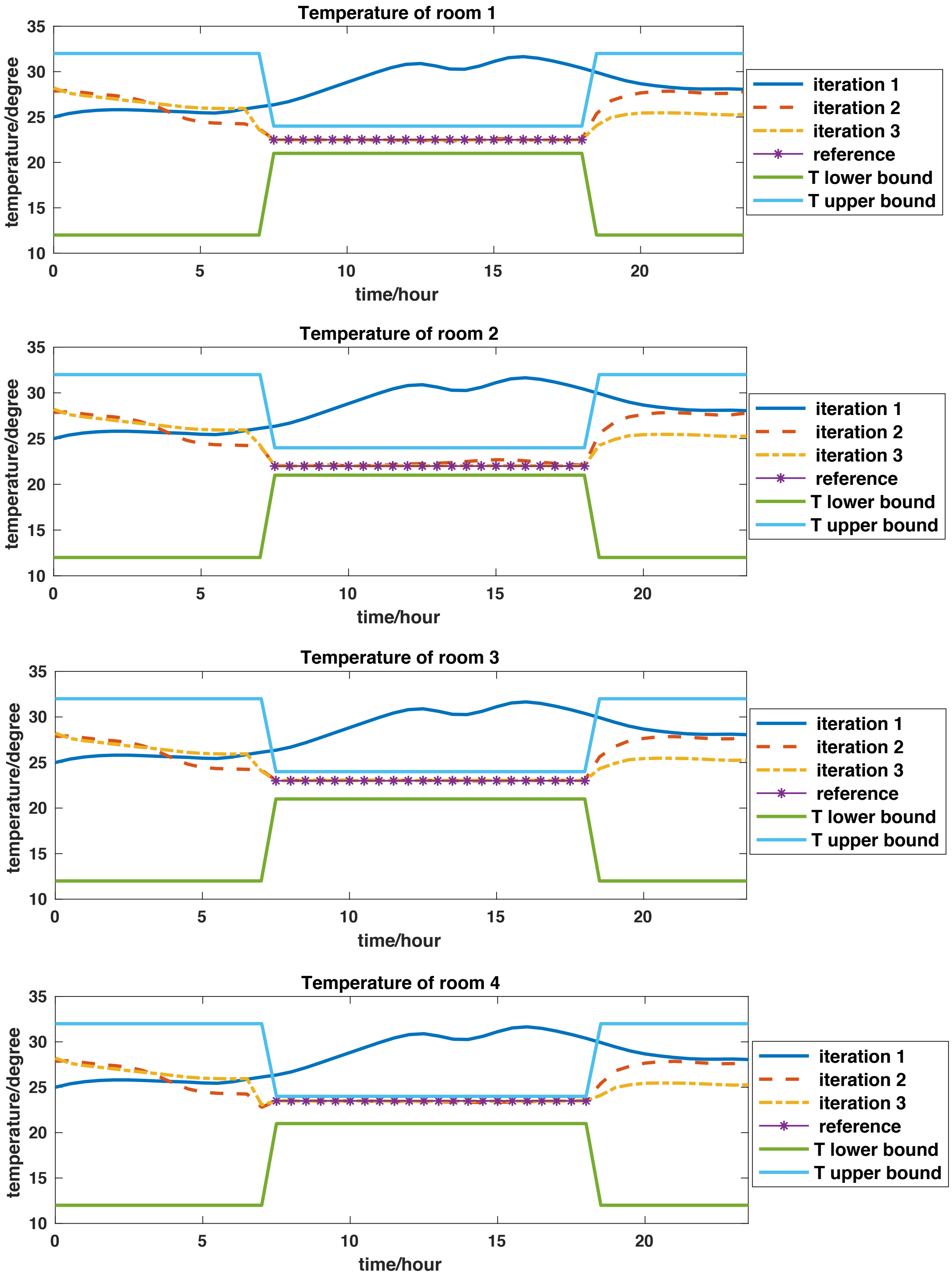

Figure 5.1: Temperature of the four zones in three iterations

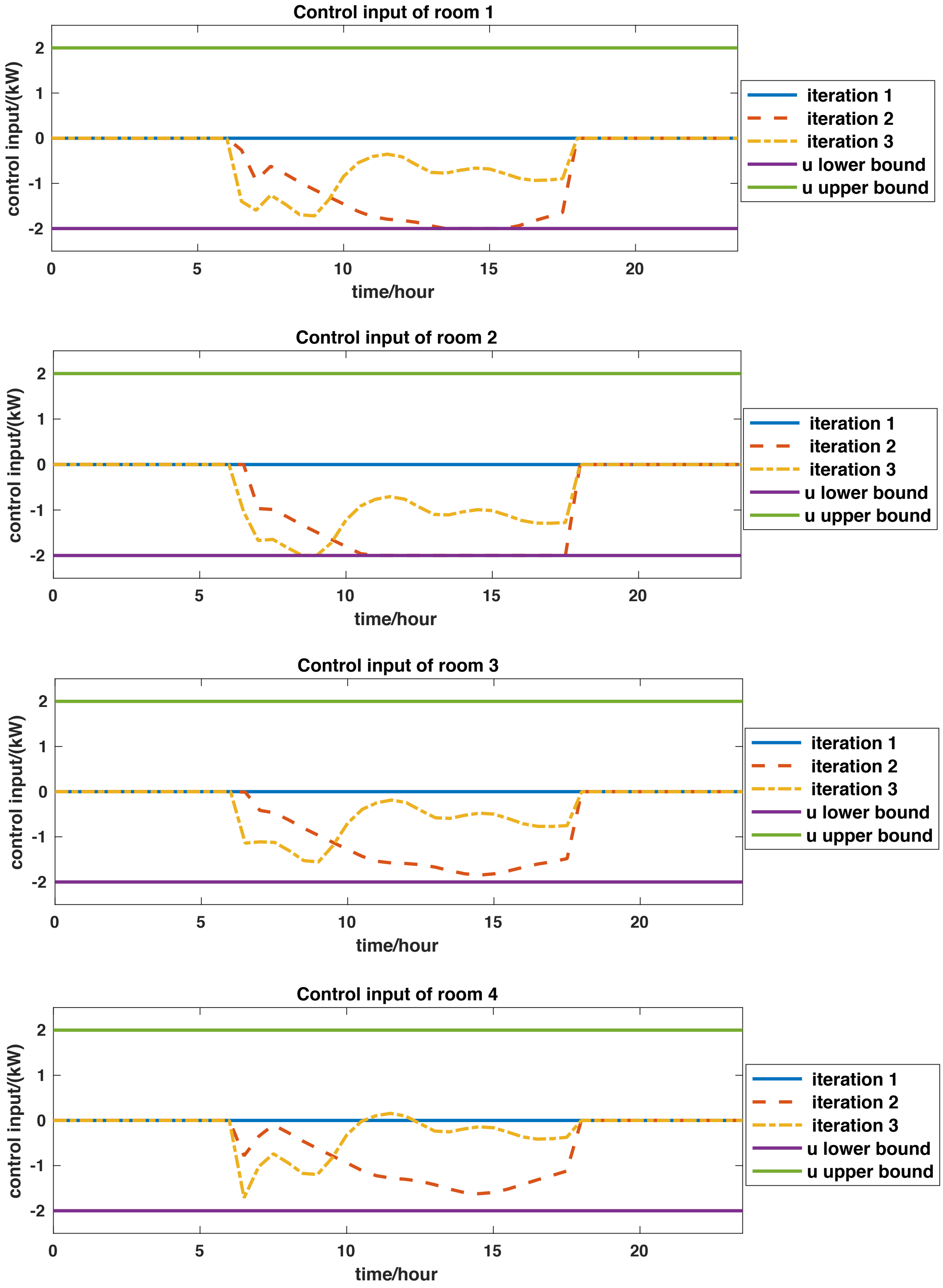

Figure 5.2: Control input of four zones in three iterations

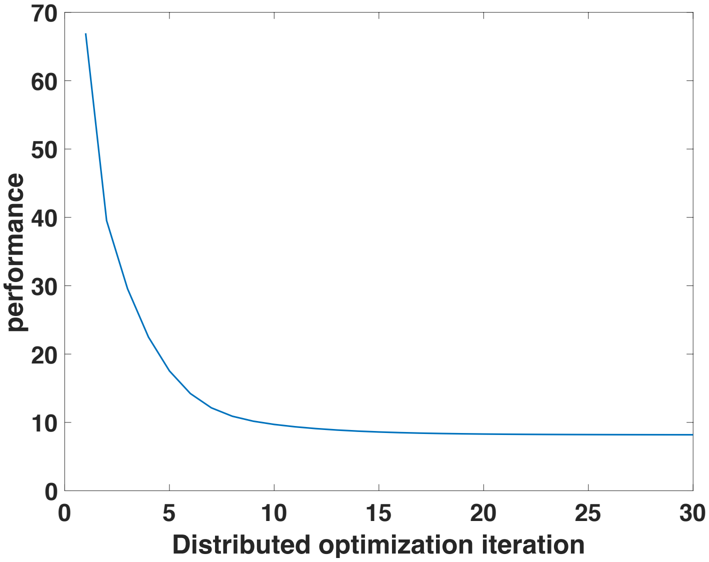

Figure 5.3: Algorithm performance in distributed optimization iteration domain

algorithm, large-scale thermal systems can be well handled with similar achievable performance with the centralized approach but low time cost. The convergence of the algorithm has also been established.

Part II

Autonomous Driving System

Chapter 6

Cooperative Driving Based on Negotiation with Persuasion and Concession

Recently, along with emergence of autonomous driving vehicles, it is predicted that in the near future, human drivers need to share road and interact with self-driving cars in all the possible traffic scenarios. In order to achieve smooth interaction results similar with human drivers, the ego autonomous vehicle's behavior needs to be controlled intelligently. However, since the other traffic participants' information is not accessible to the ego autonomous vehicle, the centralized and distributed control approach cannot be applied. Therefore, the decentralized control method with properly designed information gathering strategy becomes the most promising candidate.

In this chapter, an algorithm based on negotiation with both persuasion and concession is proposed to tackle the challenging task of cooperative driving involving both human and robot drivers. The decision making process is formulated as an optimization based negotiation problem. The persuasion of autonomous vehicle is achieved by making commitment to tipping towards cooperation in the form of convex constraint. The concession is accomplished by gradually tuning weights in the objective function. We propose an approach suitable for most common driving scenarios including ramp merging, lane keeping/changing and intersection crossing. The effectiveness of the proposed algorithm is demonstrated by simulation for several different driving scenarios.

6.1 Introduction

With recent development of autonomous driving technology, it is becoming more and more convincing that a driving environment dominated by autonomous vehicles is coming in the future. However, current self-driving cars are not mature enough to totally replace human drivers. Several years or even decades of effort is needed to bring the conceptual fully connected autonomous driving into reality, which means that autonomous cars and human drivers will coexist for quite a long time. This predictable situation leads to the requirement that robot drivers not only need to communicate and interact with its robot siblings but also need to cooperate with human drivers safely and intelligently.

Recent works on cooperative driving mainly focus on cooperation among connected vehicles assuming that each vehicle can be controled by the same strategy [10], [18], [19]. With this assumption, the cooperative driving task can be formulated as a multi-agent system control problem. However, when it comes to human drivers who will follow his/her own driving logic or habits, the algorithms based on multi-agent system theory will fail as some of agents (human drivers) cannot be controlled and their behaviors may or may not comply with the multi-agent system's cooperation objective. In these scenarios, most algorithms simply require the autonomous vehicles to yield people or assume they are following certain simple predefined driving pattern. The disadvantages of these algorithms are obvious, the resulting autonomous vehicles will either be too conservative (bullied by human drivers in the extreme case) or drive into some dangerous situations due to incorrect assumption of human behavior.

Another approach dealing with human drivers is to model their behaviors as solutions of an optimal planner, the reward function of which can be learned through human driving data [49]. This method did produce some promising results illustraing autonomous cars' influence on human driver's action. However, this algorithm heavily depend on the accuracy of the learned reward functions. In addition, human drivers' behavior may be influenced by other road users so that reward function itself may not be reliable enough.

In this chapter, in order to enable intelligent and cooperative interaction without dependency on learned reward function, we proposed to use human driver's prediction informaition and make cooperative proposals accordingly by solving an optimization problem. Our algorithm models the decision making task as a negotiation process between human and robot (H-R) drivers similar to negotiation between human and human (H-H). Two key factors in H-H negotiation are concession and persuasion, which work together to help reach an offer acceptable to both sides [58], [24]. In the autonomous driving context, concession is formulated as sacrifice in self-objective and persuasion is fulfilled by showing preference among several available strategies. By their co-effect, the autonomous car can make its behavior more readable to human which results in more effective cooperation.

The chapter is organized as follows. In Section 6.2, the cooperative driving prolem is formulated as an optimization problem assuming all the traffic participants are autonomous vehicles. In Section 6.3, the simplistic conservative control strategy is introduced. In Section 6.4, the human driver factors are taken into account which transform the optimization problem is transferred to a negotiation process. In section 6.5, simulation results in several driving scenarios are presented to illustrate the effectiveness of the proposed algorithm's. Section 6.6 concludes the chapter.

6.2 Cooperative Driving

In this section, the mathematical description of the vehicle dynamics is introduced and an optimization problem is formulated for cooperative driving in the framework of Model Predictive Control (MPC). The system is treated as a two-agent system consisting of two autonomous cars, both of which can be controlled.

Vehicle Modeling

A popular model used for vehicle dynamics is the bicycle model [46]. We adopted the discretized version of the bicycle model described by the following difference equations:

$$\begin{aligned} x_{k+1} &= x_k + T_s v_k \cos\left(\theta_k + \tan^{-1}\left(\frac{L_r}{L}\tan\delta_k\right)\right), && (6.1)\\ y_{k+1} &= y_k + T_s v_k \sin\left(\theta_k + \tan^{-1}\left(\frac{L_r}{L}\tan\delta_k\right)\right), && (6.2)\\ \theta_{k+1} &= \theta_k + T_s v_k \frac{\tan\delta_k}{L}\cos\left(\tan^{-1}\left(\frac{L_r}{L}\tan\delta_k\right)\right), && (6.3)\\ v_{k+1} &= v_k + T_s a_k, && (6.4) \end{aligned}$$

where k is the time index, T_s is the sampling time, the state vector $[x, y, \theta, v]^T \in \mathbb{R}^{4\times 1}$ includes mass center x, y positions, yaw angle and speed of the vehicle in the road aligned coordinate frame respectively. Control input vector of the system $[a, \delta]^T \in \mathbb{R}^{2\times 1}$ consists of the vehicle's acceleration a and steering angle δ. L_r, L_f and $L = L_r + L_f$ are the rear, front and full length of the car respectively.

Compactly, (6.1)-(6.4) can be written as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \tag{6.5}$$

where $\mathbf{x}_k = [x_k, y_k, \theta_k, v_k]^T$ and $\mathbf{u}_k = [a_k, \delta_k]^T$ are the state and control input vector respectively.

Optimization Formulation

In this subsection, the two-agent cooperative driving problem is formulated as an optimization problem applying the bicycle model.

Aiming at achieving driving efficiency and satisfying comfort demand, the following cost function is defined:

$$J_k = \sum_{i=1}^{2} \frac{1}{Z_i} \sum_{t=0}^{N-1} \left(\|\mathbf{x}_{i,k+t|k} - \mathbf{x}_{des}\|_Q^2 + \|\Delta\mathbf{u}_{i,k+t|k}\|_R^2 \right), \tag{6.6}$$

where k is the current time instant, i is the vehicle index, N is the receding horizon length, Z_i are predefined normalizing factors and $\mathbf{x}_{des}$ is the desired state. $\mathbf{x}_{i,k+t|k}$ and $\mathbf{u}_{i,k+t|k}$ denote state and control input signal at $k+t$ predicted at k starting from the current state. $\Delta\mathbf{u}_{i,k+t|k} = \mathbf{u}_{i,k+t|k} - \mathbf{u}_{i,k+t-1|k}$ defines the control change, representing the comfort level, Q and R are penalty matrices for deviation from desired state and control change respectively.

With regard to the constraints, the most critical one is associated with safety. To guarantee collision avoidance with both moving obstacles i.e. other vehicles sharing the road and static ones such as lane bounds and parked vehicles, the following safety constraints are defined:

$$\begin{aligned} & y_{min,k+t} \leq y^j_{i,k+t|k} \leq y_{max,k+t}, && (6.7)\\ & (x_{1,k+t}, y_{1,k+t}) \in \mathbb{S}(x_{2,k+t}, y_{2,k+t}), && (6.8)\\ & i \in \{1,2\}, \quad j \in \{1,2,3,4\}, \quad t = 0,1,\cdots,N-1, \end{aligned}$$

where (6.7) describes constraints formulated by static obstacles and (6.8) accounts for safety constraint between two agents. y^j's denote the lateral positions of the vehicle's four corners, where

$$\begin{aligned}
\text{front right}: \quad & y^1 = y + L_f \sin\theta - \tfrac{W}{2}\cos\theta, \\
\text{front left}: \quad & y^2 = y + L_f \sin\theta + \tfrac{W}{2}\cos\theta, \\
\text{rear right}: \quad & y^3 = y - L_r \sin\theta - \tfrac{W}{2}\cos\theta, \\
\text{rear left}: \quad & y^4 = y - L_r \sin\theta + \tfrac{W}{2}\cos\theta,
\end{aligned}$$

and W is the width of the vehicle. y_{min} and y_{max} denote lower and upper bounds of lateral position defined by the lane and parked cars and $\mathbb{S}(x,y)$ represents the safety set formulated by position (x,y). There exist quite a few approaches to define safety set $\mathbb{S}(x,y)$ in literatures. One simple case exemplifying $\mathbb{S}(x,y)$ is the halfplane defined by the line passing through (x,y) with slope equal to $\tan\theta$. Another example is the region outside a circle centered at (x,y) with radius of pre-assigned safety distance.

In addition to the safety constraints, other demands imposing on the vehicle consist of the initial state consistency, control saturation, kinematics and dynamics feasibility which are formulated as:

$$\begin{aligned}
& \mathbf{u}_{i,k+t|k} \in \mathbb{U}_i, && (6.9) \\
& \mathbf{x}_{i,k+t+1|k} = f(\mathbf{x}_{i,k+t|k}, \mathbf{u}_{i,k+t|k}), && (6.10) \\
& \mathbf{x}_{i,k|k} = \mathbf{x}_i(k), && (6.11) \\
& i \in \{1,2\} \quad t = 0,1,\cdots,N-1,
\end{aligned}$$

where $\mathbb{U}_i$ is the feasible set of vehicle i's control input $\mathbf{u}_i$ and $\mathbf{x}_i(k)$ is vehicle i's current state.

Combining cost function (6.6) and constraints (6.7)-(6.11), the agent-based cooperative driving optimization problem can be stated as:

$$\begin{aligned}
\min_{\mathbf{u}_{i,k+t|k}, i=1,2, t=0,1,\cdots,N-1} \quad & J_k && (6.12) \\
s.t. \quad & \text{constraints}(6.7)-(6.11)
\end{aligned}$$

Note that in this optimization, both the agents can be controlled so that the Pareto optimal solution can be achieved in which no vehicle can obtain better performace without hurting the other one. This Pareto optimal solution is defined to be the cooperative driving strategy.

6.3 Conservative Driving Strategy

The most intuitive approach to handle human factors in driving is to assign human drivers with the highest priority. As a consequence, autonomous vehicles are required to always yield human drivers who can take any trajectory they please. Mathematically, this is equal to setting human drivers' behavior as hard constraints, which adjusts the original optimization problem (6.12) to the following:

$$
\begin{aligned}
\min_{\mathbf{u}_{1,k+t|k}} \quad & \sum_{t=0}^{N-1} \left(\|\mathbf{x}_{1,k+t|k} - \mathbf{x}_{des}\|_Q^2 + \|\Delta \mathbf{u}_{1,k+t|k}\|_R^2 \right) \qquad (6.13) \\
s.t. \quad & y_{min,k+t} \leq y^j_{1,k+t|k} \leq y_{max,k+t}, \\
& (x_{1,k+t|k}, y_{1,k+t|k}) \in \mathbb{S}(x_2^p(k+t), y_2^p(k+t)), \\
& \mathbf{u}_{1,k+t|k} \in \mathbb{U}_1, \\
& \mathbf{x}_{1,k+t+1|k} = f(\mathbf{x}_{1,k+t|k}, \mathbf{u}_{1,k+t|k}), \\
& \mathbf{x}_{1,k|k} = \mathbf{x}_1(k), \\
& j \in \{1,2,3,4\}, \quad t = 0, 1, \cdots, N-1,
\end{aligned}
$$

where the autonomous car and human driven car are assigned to be vehicle 1 and 2 respectively without loss of generality. $(x_2^p(k+t), y_2^p(k+t))$ is the prediction of human driver's future trajecotry, which is assumed to be accessible either by an extra prediction module or Vehicle-to-Vehicle (V2V) communication. Compared to optimization (6.12), decision variables in the altered optimiztion (6.13) only involve the autonomous car.

Note that as the autonomous vehicle always treats human driver's behavior as hard constraints, the human driver thus has no incentive to change his planned trajectory. Resultantly, the solution to (6.12) is a conservative instead of a cooperative one for human driver's behavior cannot be influenced. Hence there is no interaction between vehicles. The outcome of this conservativity can be easily foreseen. An examplar situation is that an autonomous vehicle and a vehicle driven by human meet at an intersection. In this scenario, the autonomous car will always let the human driver pass first regardless of whether this decision is reasonable or not. In the extreme case where traffic is heavy on the other lane and all human drivers happen to be selfish implying that they will not yield the autonomous car spontaneously, the robot-driver then has no choice but a ridiculously long wait.

6.4 Human Robot Driver Negotiation

In this section, the existence of human drivers in the traffic system is considered. The original agent-based optimization problem is extended to a negotiation process where the autonomous driving vehicle attempts to "talk" with human drivers by making cooperative proposals and modifying them according to human drivers' corresponding behaviors.

In order to avoid undesirable over-cautious behavior caused by the conservative algorithm in (6.13), we formulate the cooperative drving problem as a negotiation process with concession and persuasion reflecting human's natural behavior in negotiation. The concession is fulfilled by introducing a tunable cost function and the persuasion is achieved by integrating a persuasion constraint into the original optimization (6.13).

Negotiatoin has been an effective tool in almost every aspect of the society. People are involved in different forms of negotiation purposely or unconciously. Namely, in the business field, companies are vigorously negotiating with other companies or customers for product price and material costs. In the politics area, each nation negotiates with the others for their own benefits. In the traffic system, human drivers are also negotiating with others for right of the road in whatever driving scenarios. A general definition of negotiation is the necessary dialogues between agents in order to arrive at a globally acceptable offer when two or more of them have conflicting preferences over a set of issues or a fixed amount of resource. In addition, agents are most likely to be only aware of its own preference without enough knowledge about the other participants in most such negotiations. This feature requires the players to make tentative proposals and check for the others' reactions. In the specific driving problem, this proposal-based behavior can be observed in all common scenarios:

Lane Changing

Suppose a rantional human driver intending to change to the neighboring lane, he/she will first turn on the indicator, then slightly move towards bound of the target lane in order to make the behavior more evident and eventually finish lane changing when a safe space is existent. This safe gap can be found by either patient waiting or persuading the car behind him/her on the target lane to decelerate for driving efficiency purpose. The indicator and possible persuasion process both constitute the proposals in lane changing case.

Lane Keeping

Now we switch the point of view from drivers making lane changing proposals to drivers already on the target lane but passively influenced by the lane changing cars. To obviously show his/her preference for yielding (not yielding), the lane keeping cars are predicted to decelerate (accelerate or keep current speed). These possible actions formulate the lane keeping cars' proposals representing their responses to lane changing cars'.

Ramp Merging

Ramp merging scenario is quite similar to the lane changing case except that it has to be completed in a certain distance and there is a speed difference between self-car and the target lane cars. The proposals in this case are still defined by indicator and turning behavior.

Intersection Crossing

Suppose two cars reach an intersection without stop signs around the same time, then they need to negotiate about who should pass first. Another interesting situation is the two-way stop (cross traffic does not stop), the question facing the drivers stopping at the stop sign is how to find or make a safe space for passing. In these examples, drivers again have two choice, which are passing as soon as possible and waiting patiently. The corresponding proposals are then moving forward to show preference for passing and staying still or even backing up to express patience.

In the following subsection, the negotiation based cooperative driving problem is formulated as an optimization.

Negotiation Formulation

In order to enable negotiation between robot driver and human driver, the control input of both the human driver vehicle and the autonomous vehicle should be treated as decision variables. Meanwhile, the prediction of human driving behavior should be taken into account. Hence, the negotiation objective function is adjusted as

$$\begin{aligned} J_k^n &= \frac{1}{Z_1}\sum_{t=0}^{N-1}\left(\|\mathbf{x}_{1,k+t|k}-\mathbf{x}_{1,des}\|_Q^2+\|\Delta\mathbf{u}_{i,k+t|k}\|_R^2\right) \\ &+\frac{1}{Z_2}\sum_{t=T_n}^{N-1}\|\mathbf{x}_{2,k+t|k}-\mathbf{x}_2(k+t)^p\|_2, \end{aligned} \tag{6.14}$$

where $\mathbf{x}_2(k+t)^p$ is the human driver's state trajectory prediction and T_n is the pre-defined near future horizon during which human driver is assumed to follow the prediction exactly. The first term in (6.14) is the same as conservative objective function in (6.13) and the second term is included to reflect human driver's intention. Given that autonomous vehilce presumably has no access to accurate prior knowledge about human driver's objective function, it is assumed that the prediction trajectory minimizes (maximizes) human driver's hidden cost (reward) function. Thus, via minimizing the difference between planned human driver trajectory and predicted trajecotry in the 2-norm sense, the autonomous vehicle is enabled to consider its possible intervention to the human driver in its planning step.

The constraints for the negotiation optimziation problem will be altered correspondingly as:

$$\begin{aligned} & y_{min,k+t}\le y_{i,k+t|k}^j\le y_{max,k+t}, \quad j\in\{1,2,3,4\} \\ & \mathbf{u}_{i,k+t|k}\in\mathbb{U}_i, \quad \mathbf{x}_{1,k|k}=\mathbf{x}_1(k), \\ & \mathbf{x}_{i,k+t+1|k}=f(\mathbf{x}_{i,k+t|k},\mathbf{u}_{i,k+t}), \\ & t=0,1,\cdots,N-1, \text{ if } i=1, \\ & t=T_n,T_n+1,\cdots,N-1, \text{ if } i=2, \\ & (x_{1,k+t|k},y_{1,k+t|k})\in\mathbb{S}(x_2^p(k+t),y_2^p(k+t)), \\ & t=0,1,\cdots,N-1, \\ & \mathbf{x}_{2,k+t|k}=\mathbf{x}_2(k+t)^p, \quad t=0,\cdots,T_n-1 \end{aligned} \tag{6.15}$$

Summarily, the negotiation problem is formulated as:

$$\min_{\substack{\mathbf{u}_{1,k+t},t=0,1,\cdots,N-1\\ \mathbf{u}_{2,k+t},t=T_n,\cdots,N-1}} \quad J_k^n \tag{6.16}$$

$$s.t. \quad \text{set of constraints}(6.15).$$

Remark 1: The solution to the optimzation problem (6.16) is not broadcasted to the human driver so that vehicle-to-vehicle (V2V) communication is not necessary in this approach. The autonomous vehicle will take the action given by the solution showing its intention. In this manner, the human-robot driver interaction can be fulfilled.

Remark 2: Despite treating the human driver's control input as one of optimziation decision variables, it does not mean that human driver will follow the solution obtained. The planned trajectory merely serves as a proposal to the human driver which may be accepted or refused.

However, the interaction modeling still remains incomplete as human driver's response is not included yet. Thus, we propose the introduction of two more factors which are concession and persuasion strategies in the next subsections.

Concession Strategy

The most intuitive and natural strategy when negotiating with another party possessing a different utility function is to make concession, revising the originally proposed relatively more selfish offer to a more likely acceptable one for both sides. This idea has been formalised as a negotiation protocol called monotonic concession protocol [13]. A negotiation protocol is defined as a set of rules or sequential steps restricting negotiating agents' available moves into a certain range. Specifically, the monotonic concession protocol requires the agent's desired utility being monotonically decreasing.

However, negotiation in driving context not only depends on the cost function but also on the constraints imposed by environment and other agents. Moreover, each move in the driving negotiation means an actual move in the real world which is far more safe critical compared to traditional negotiations encountered in business field. Hence, the simplistic monotonic concession strategy is not suitable for the autonomous driving problem.

In order to fit the concession characteristic into the optimization based negotiation problem defined by (6.16), a time-varying tunable weight is introduced to the cost fucntion. The adjusted cost function is as following:

$$\begin{aligned} J_k^{n,c} &= \frac{w_k}{Z_1}\sum_{t=0}^{N-1}\left(\|\mathbf{x}_{1,k+t|k}-\mathbf{x}_{1,des}\|_Q^2+\|\Delta\mathbf{u}_{i,k+t|k}\|_R^2\right) \\ &\quad +\frac{1-w_k}{Z_2}\sum_{t=T_n}^{N-1}\|\mathbf{x}_{2,k+t|k}-\mathbf{x}_2(k+t)^p\|_2. \end{aligned} \tag{6.17}$$

The tunable weight w_k at time instant k is defined as

$$\begin{cases} w_0 = 0.5, \\ w_{k+1} = w_k\beta^{1-p_{a,k}}, \end{cases} \tag{6.18}$$

where $\beta<1$ is a predefined scalar and $p_{a,k}$ is the proposal acceptance probability at time instant k reflecting intention of the human driver involved in the negotiation. The initial value w_0 is set to 0.5 for at the first round of negotiation, it is assumed that autonomous vehicle and human driver vehicle share equal priority using the road.

As the negotiation proceeds, the priority of human and autonomous vehicle will be adjusted according to human driver's acceptance of proposal. Particularly, if human driver instantly accepts the proposed trajectory by exactly following it thereafter, then $p_{a,k}$ will be equal to 1 so that w_k remains at 0.5, assigning autonomous vehicle and human driver the same priority for the entire negotiation process. Another extreme case is that human driver totally refuses to consider the proposed driving plan, never willing to change his/her originally planned trajectory and act in ignorance of existence of the autonomous vehicle. In this scenario, $p_{a,k}$ will quickly decrease to nearly 0 and the autonomous vehicle's behavior will coincide with the conservative strategy as human driver purposely leaves no space for negotiation. In practice, these two extremes may be quite rare, most probably $p_{a,k}$ will be some number between 0 and 1 and keep changing during the negotiation process as human driver's behavior may not properly reflect his/her true intention or the human drivers simply changes his/her intention which is a fairly common phenomenon in everyday life.

Therefore, definition of the acceptance probability term $p_{a,k}$ is crucial to the algorithm. In order to model human drivers' reaction in the negotiation, we propose to define $p_{a,k}$ as

$$p_{a,k} = \frac{\|trj_k^p - trj_{k-1}^p\|_2^2}{\|trj_k^p - trj_{k-1}^p\|_2^2 + \|trj_k^p - trj_{k-1}^{h,c}\|_2^2}, \tag{6.19}$$

where

$$\begin{aligned} trj_k^p &= [x_{2,k|k}^p, \cdots, x_{2,k+t|k}^p, \cdots, x_{2,k+T-1|k}^p, \\ &\quad y_{2,k|k}^p, \cdots, y_{2,k+t|k}^p, \cdots, y_{2,k+T-1|k}^p]^T \end{aligned}$$

represents the predicted (x,y) trajectory of the human driver vehicle at time instant k. Similarly,

$$\begin{aligned} trj_{k-1}^p &= [x_{2,k|k-1}^p, \cdots, x_{2,k+t|k-1}^p, \cdots, x_{2,k+T-1|k-1}^p, \\ &\quad y_{2,k|k-1}^p, \cdots, y_{2,k+t|k-1}^p, \cdots, y_{2,k+T-1|k-1}^p]^T \end{aligned}$$

denotes the prediction at the last time instant which is human driver's original plan before the autonomous vehicle makes the move and

$$\begin{aligned} trj_{k-1}^{h,c} &= [x_{2,k|k-1}, \cdots, x_{2,k+t|k-1}, \cdots, x_{2,k+T-1|k-1}, \\ &\quad y_{2,k|k-1}, \cdots, y_{2,k+t|k-1}, \cdots, y_{2,k+T-1|k-1}]^T \end{aligned}$$

is the solution trajectory for human to the negotiation optimization problem (6.16) at the last time step $k-1$, i.e., the previous cooperative driving proposal for human driver.

The definition (6.19) is based on the idea that the human driver is given two options at each decision making step. One is to follow his/her original plan and the other one is to cooperate with the autonomous car after recognizing its intention leading to a predicted trajectory relatively close to the cooperative porposal. Thus, the similarity of the new predicted trajectory with the initial planned one and the cooperative proposal provides us with a metric of acceptance probability. There are multiple ways to describe similarity between two trajectories such as the simplest Euclidean distance and more complex ones including Fréchet distance, dynamic time warping (DTW), longest common subsequence (LCSS) and so on [54]. In this dissertation, the Euclidean distance (2-norm) is selected due to the same sampling rate of trajectories, intuitive physical meaning and its simplicity. Therefore, the acceptance probability (6.19) can be interpreted as normalized trajectory similarity between updated and original prediction, inferring that less similar the new prediction is with the original one, the larger p_a is and more likely human driver is to cooperate.

Although with the tunable concession weight, the autonomous vehicle is capable of interacting with other road users more efficiently than treating human prediction as hard constraints, it is still quite conservative as concession itself can only express yielding intention. Therefore, in the next subsection, the persuasion factor is introduced to assist the autonomous vehicle in making decisions.

Persuasion Strategy

In order to distinguish the negotiating agents from ones with conservative strategies, in addition to concession, the autonomous vehicle also needs to express its intent to cooperate properly, claiming its own right of using the raod. As concession strategy, persuasion is another useful technique in negotaition. Unlike concession focusing on making more acceptable offers, persuasion on the other hand is utilized to influence opponents' beliefs and behaviors so that the offer can be more beneficial to the self-agent. The persuasion used in business and other fields where interaction happens is mainly based on dialog. Dialogs enable agents holding different points of view or involved in a conflict of interest to exchange arguments and persuade each other [48]. However, in the context of driving, as all the agents are making moves simultaneously and the dialogs are in the form of driving behavior, it is necessary to reformulate the concrete implementation of persuasion.

In this dissertation, the persuasion is introduced as a constraint imposed on the negotiation process formulated in the following:

$$p_{coop,k} \geq \alpha_k p_{con,k}, \tag{6.20}$$

where $\alpha_k{=}1{+}\gamma p_{a,k}$ is a scalar dependent on $p_{a,k}$ defined by (6.19) and a predefined parameter γ, $p_{coop,k}$ and $p_{con,k}$ denote the cooperating and conservative probability at time instant k respectively. Following the same idea of definition of $p_{a,k}$, $p_{coop,k}$ and $p_{con,k}$ are formulated as:

$$p_{coop,k} = \frac{\|trj_k^{coop} - trj_{k-1}^{con}\|_2^2}{\|trj_k^{coop} - trj_{k-1}^{con}\|_2^2 + \|trj_k^{coop} - trj_{k-1}^{coop}\|_2^2}, \tag{6.21}$$

$$p_{con,k} = \frac{\|trj_k^{coop} - trj_{k-1}^{coop}\|_2^2}{\|trj_k^{coop} - trj_{k-1}^{con}\|_2^2 + \|trj_k^{coop} - trj_{k-1}^{coop}\|_2^2}, \tag{6.22}$$

where

$$\begin{aligned} trj_k^{coop} &= [x_{1,k|k}, \cdots, x_{1,k+t|k}, \cdots, x_{1,k+T-1|k}, \\ &\quad y_{1,k|k}, \cdots, y_{1,k+t|k}, \cdots, y_{1,k+T-1|k}]^T, \end{aligned}$$

is the autonomous vehicle trjectory to be solved at current time instant,

$$\begin{aligned} trj_{k-1}^{coop} &= [x_{1,k|k-1},\cdots,x_{1,k+t|k-1},\cdots,x_{1,k+T-1|k-1}, \\ &\quad y_{1,k|k-1},\cdots,y_{1,k+t|k-1},\cdots,y_{1,k+T-1|k-1}]^T, \end{aligned}$$

is the robot driver's solution to the negotiation problem at the last time step and trj_{k-1}^{con} is the conservative trajectory plan for the autonomous vehicle at previous time instant. There exist many candidates for the conservative plan trj_{k-1}^{con}, among which the most simple one is to decelerate until the interacting human vehicle pass through. A more reasonable and decent option is to use the solution from a conservative MPC like the one in (6.13). In this dissertation, we selected the latter method.

According to definition (6.21) and (6.22), it is shown that the persuasion constraint (6.20) pushes the solution trajectory to the cooperative direction harder when interacting with human drivers more likely to cooperate, i.e. higher acceptance probability.

The constraint (6.20) can be reformulated as:

$$\|trj_k^{coop} - trj_{k-1}^{con}\|_2^2 \geq \alpha\|trj_k^{coop} - trj_{k-1}^{t}rj_{k-1}^{coop}\|_2, \tag{6.23}$$

which is a convex quadratic constraint. Thus, it will not bring in much more computational load to the original problem.

The complete negotiation optimization with concession weight in the cost function and persuasion constraint is then written as:

$$\begin{aligned} \min_{\substack{\mathbf{u}_{1,k+t},t=0,1,\cdots,N-1 \\ \mathbf{u}_{2,k+t},t=T_n,\cdots,N-1}} \quad & J_k^{n,c} \qquad (6.24) \\ s.t. \quad & \text{set of constraints}(6.15), \\ & \text{persuasion constraint (6.23).} \end{aligned}$$

6.5 Simulation Results

In this section, simulations are performed in several driving scenarios based on the proposed negotiation optimization approach. The exemplifying scenarios include lane changing, lane keeping and intersection crossing. One interacting vehicle is considered in the intersection and lane keeping case, while two are taken into account for lane changing. The MPC strategies with different weights on safety are utilized to represent human drivers with different driving styles. Higher safety term is used for "nice" human drivers who is more willing to negotiate or cooperate and lower/zero weight on safety is used for selfish human drivers who are "iron nerved" and tough in negotiation. Thus, the simulation results are able to show the proposed algorithm's performance in different scenarios interacting with different kinds of human drivers.

Simulation Setting

The simulation environment utililzed in this chapter is a 1/10 scaled version of the real world and all the vehicles are assumed to be of the same size. The dimension parameters are shown in Table 6.1, where W_V and W_L are width of the vehicle and the lane respectively. The other optimization configurations are shown in Table 6.2, where $\underline{u}^e_\delta$, $\overline{u}^e_\delta$ are lower and upper bound for steering angle and $\underline{u}^e_a$, $\overline{u}^e_a$ are lower and upper bound for acceleration respectively.

Table 6.1: Dimension parameters

L_f	L_r	W_V	W_L
0.21(m)	0.19(m)	0.19(m)	0.37(m)

Table 6.2: Optimization configurations

T_s	N	$\underline{u}^e_a$	$\underline{u}^e_\delta$
0.1(s)	30	-1(m/s^2)	$-\pi/3$(rad)
$\overline{u}^e_a$	$\overline{u}^e_\delta$	y_{min}	y_{max}
1(m/s^2)	$\pi/3$(rad)	0(m)	0.74(m)
v_{min}	v_{max}	a_s	b_s
0(m/s)	1.5(m/s)	0.75(m)	0.35(m)

Lane Changing

In this scenario, the autonomous vehicle and two human driver vehicles are interacting in the setting of two-lane highway driving. The autonomous vehicle attempt to make lane changing and human drivers will decide to yield or not based on their internal driving strategy.

The strategy we use to handle multi-vehicle scenario is as follows:

- Select the front vehicle in the neighbour lane as the interacting vehicle and solve the negotiation optimization problem (6.24) accordingly.
- Check the interacting vehicle's acceptance probability (6.19). If $p_{a,k+1} - p_{a,k} \leq \varepsilon$ (ε is a predefiend small scalar) holds for five sequential time steps, then switch the interacting vehicle to the rear vehicle in the neighbour lane.
- Treat the previous interacting vehicle's trajectory prediction as hard constraint using similar method in the conservative control strategy described in section 6.3.

Figure 6.1 shows the case with "nice" human drivers (relatively high sfety weight), where the yellow car is the self autonomous vehicle, blue/green car is the front/rear vehicle in the neighbour lane and the red line represents the car trajectory. It is demonstrated in Fig. 6.1 that with persuasion constraint and concession cost function, the autonomous vehicle is able to cut in on the human driver when the cooperation proposal is accepted.

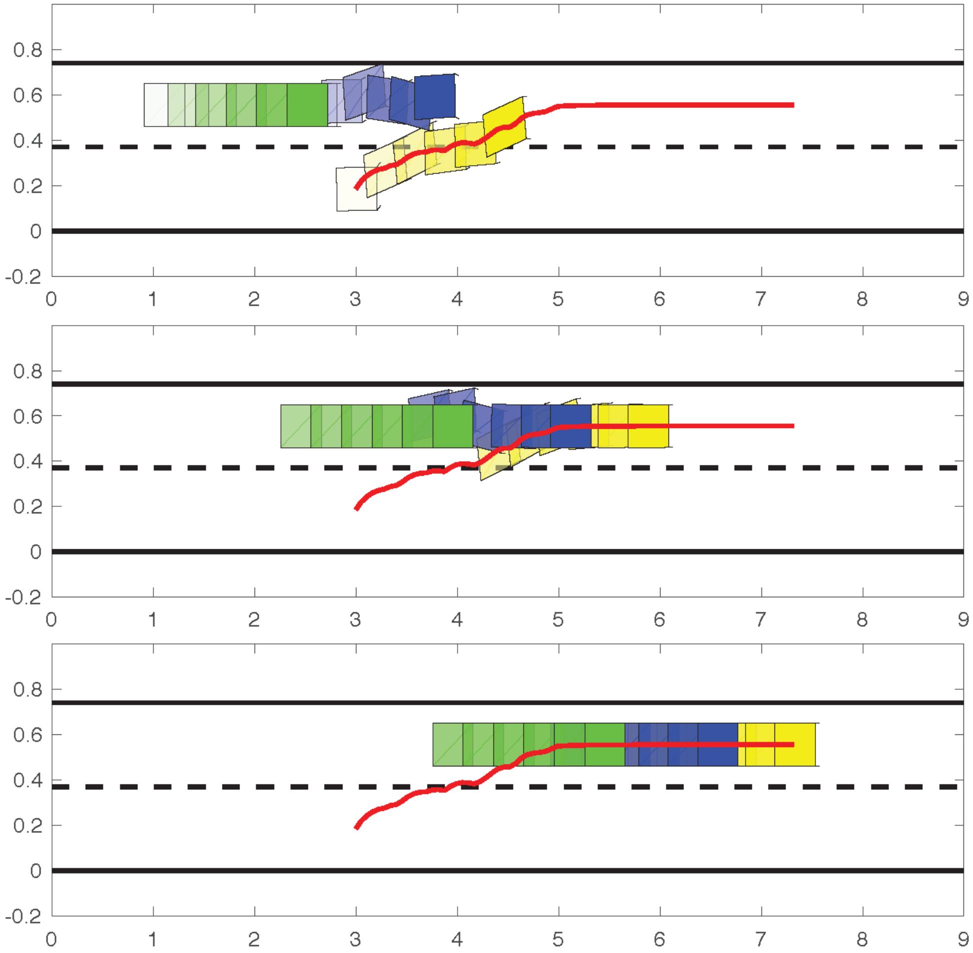

Figure 6.1: Negotiation with "nice" human driver in lane changing scenario

On the other hand, Fig. 6.2 shows the lane changing scenario interacting with a "tough" human driver (the front car) and a nice driver (the rear car). It is shown that the autonomous vehicle first attempt to cut in, however, the front tough human driver refuse to yield. The self vehicle then needs to wait and try again on the rear human driver. It turns out that the second driver is a nice one and willing to cooperate. Hence, the autonomous vehicle cut in on the rear human driver on the neighbour lane and finish lane changing.

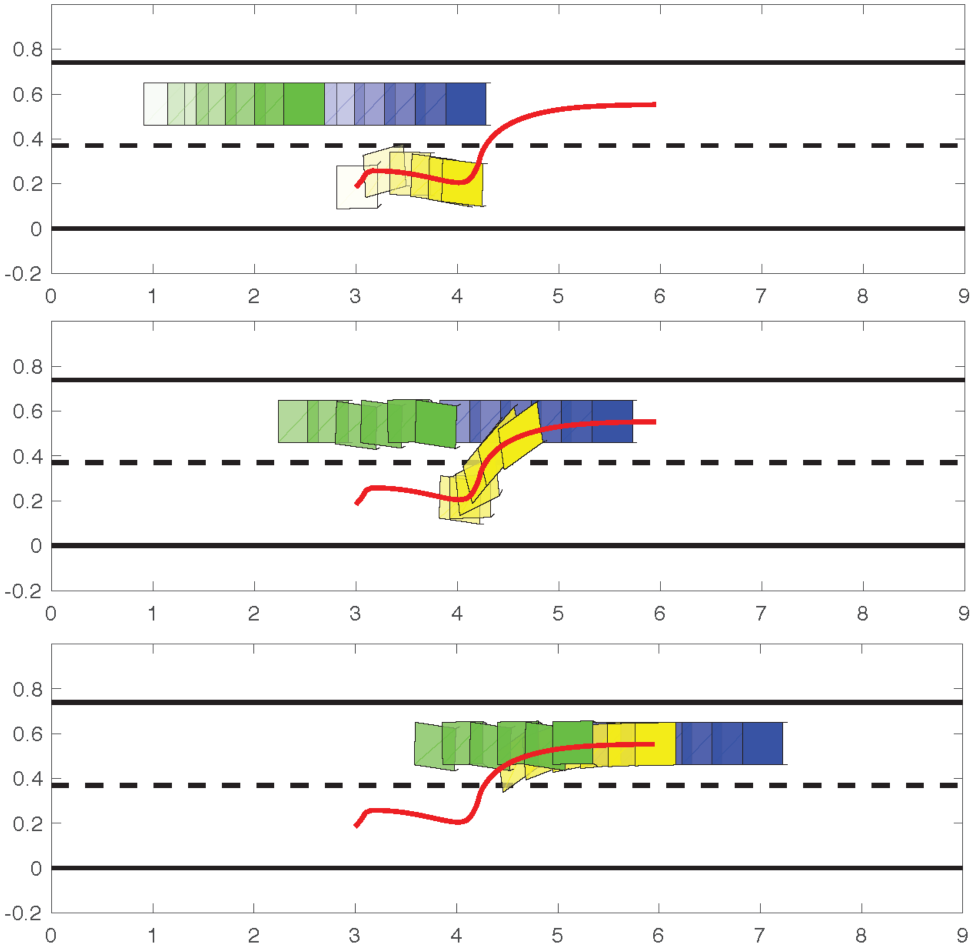

Figure 6.2: Negotiation with “tough” human driver in lane changing scenario

Lane Keeping

In this scenario, the autonomous vehicle becomes the one trying to keep lane and get to decide whether to yield the human driver who wants to make lane change. The simulation results are shown in Fig. 6.3 and Fig. 6.4 for interacting with nice and tough driver respectively, where the yellow/blue rectangle represents autonomous/human vehicle and the red solid line is the car trajectory. When interacting with a nice (tough) driver as shown in Fig. 6.3 (Fig. 6.4), the self vehicle express its intention of not yielding (yielding) by accelerating (decelerating). This effect can be seen more clearly in the speed profile plot as shown in Fig. 6.5 and Fig. 6.6.

Intersection Crossing

At the intersection, if the autonomous vehicle and the human driver vehicle arrive at the same time, the self car needs to make decision on passing first or second. The algorithm’s performance shown

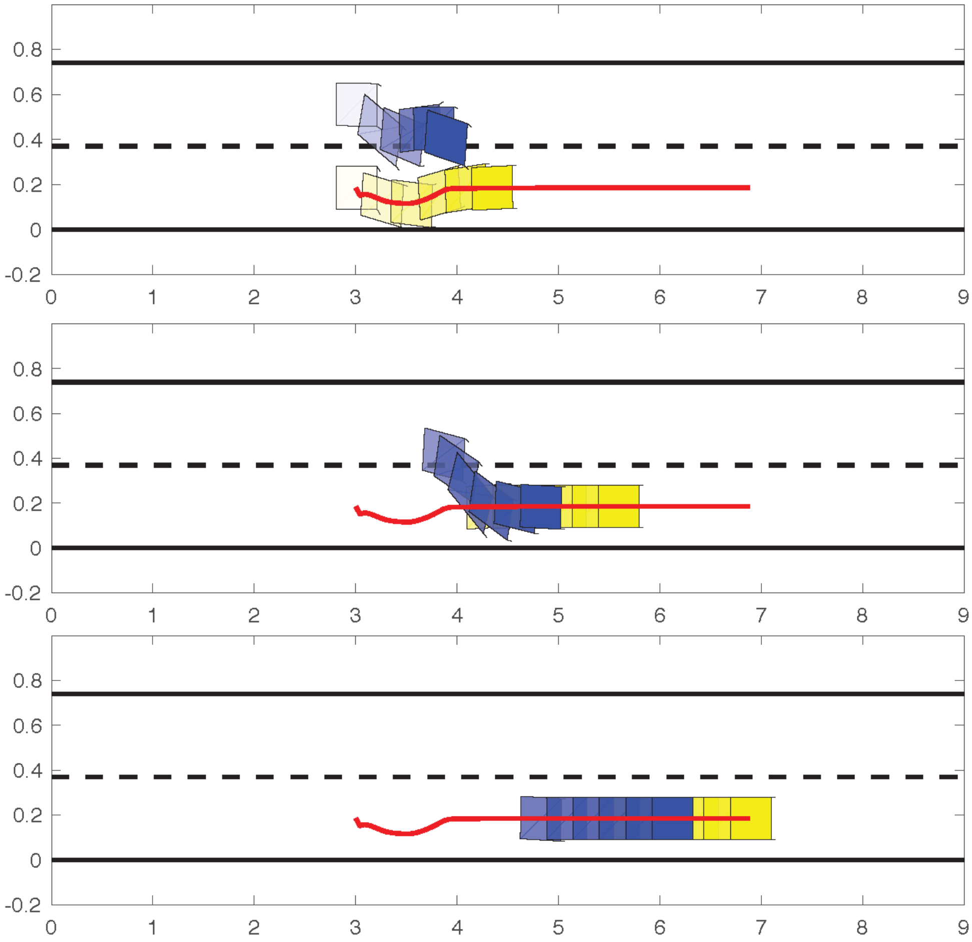

Figure 6.3: Negotiation with “nice” human driver in lane keeping scenario

in Fig. 6.7 and Fig. 6.8 demonstrate that when the other driver is nice (tough), the self vehicle will decide to cross intersection first (second). In the figures, the yellow (blue) rectangle represent self (human driver) vehicle and the red line is the car trajectory. It is shown that the algorithm leads to reasonable results consistent with human driving behavior: passing first when meeting a “nice” driver and waiting when meeting a “tough” driver. The vehicles’ speed profiles are shown in Fig. 6.9 and Fig. 6.10. These two figures illustrate that autonomous vehicle choose to accelerate and pass first when interacting with negotiable “nice” driver and keep speed low when dealing with selfish “tough” driver.

6.6 Chapter Summary

In this chapter, the cooperative driving problem was formulated as an optimization based negotiation process. For more intelligent and effective interacting among human drivers and autonomous

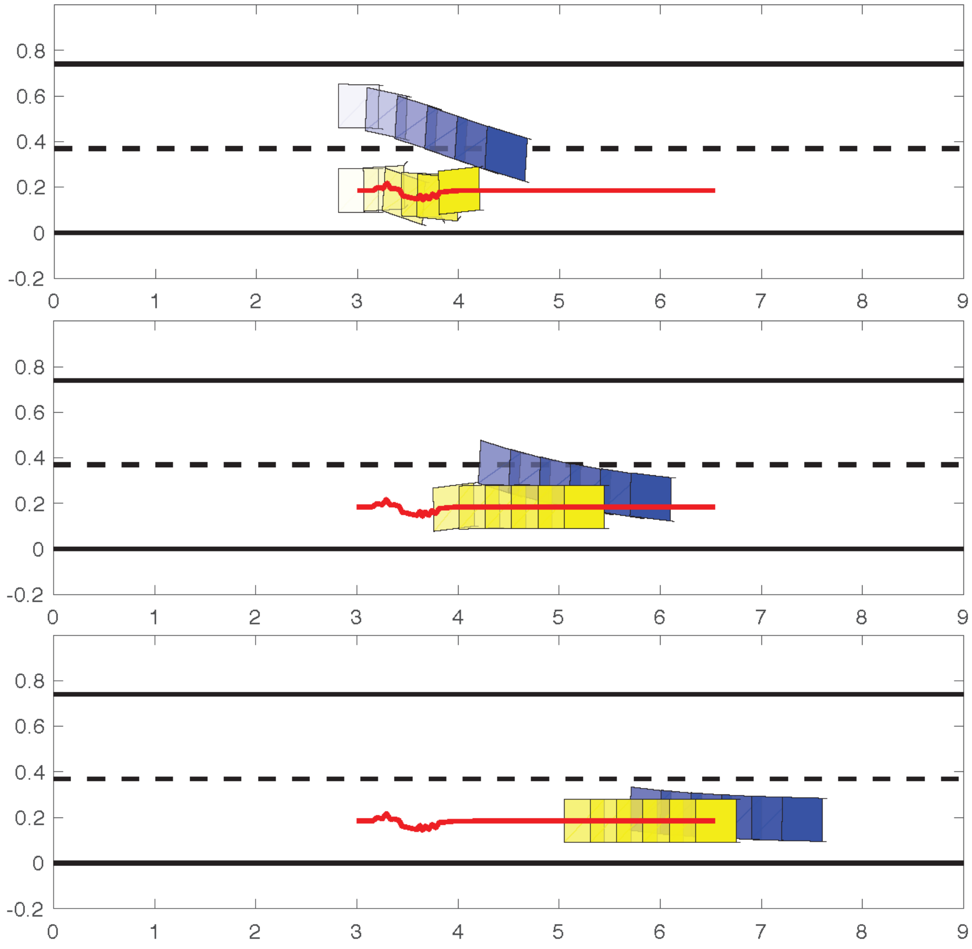

Figure 6.4: Negotiation with “tough” human driver in lane keeping scenario

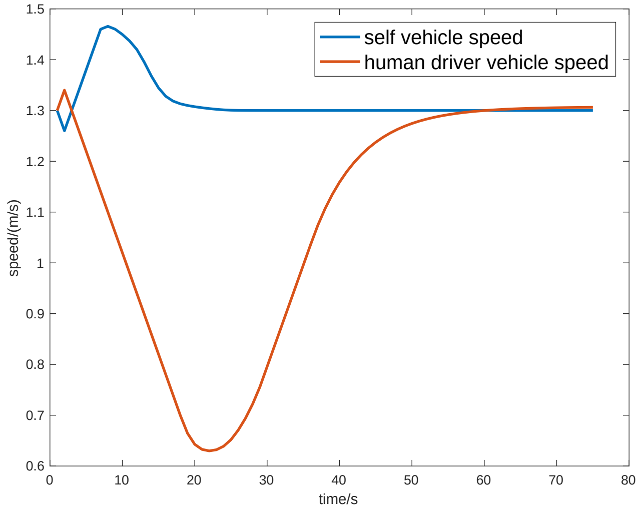

Figure 6.5: Lane keeping speed profile (negotiation with “nice” human driver case)

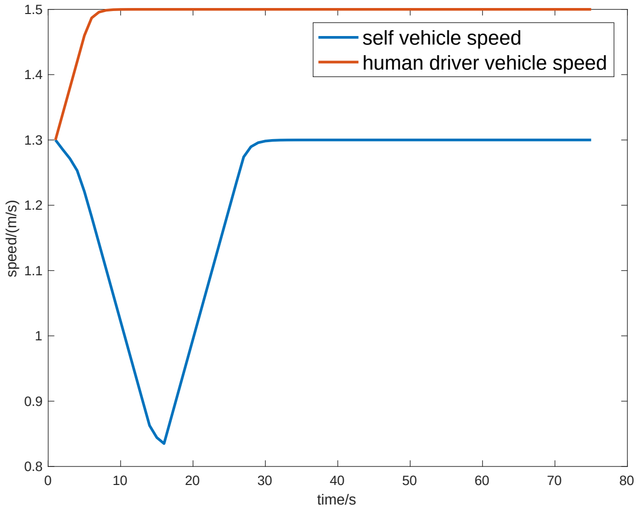

Figure 6.6: Lane keeping speed profile (negotiation with “tough” human driver case)

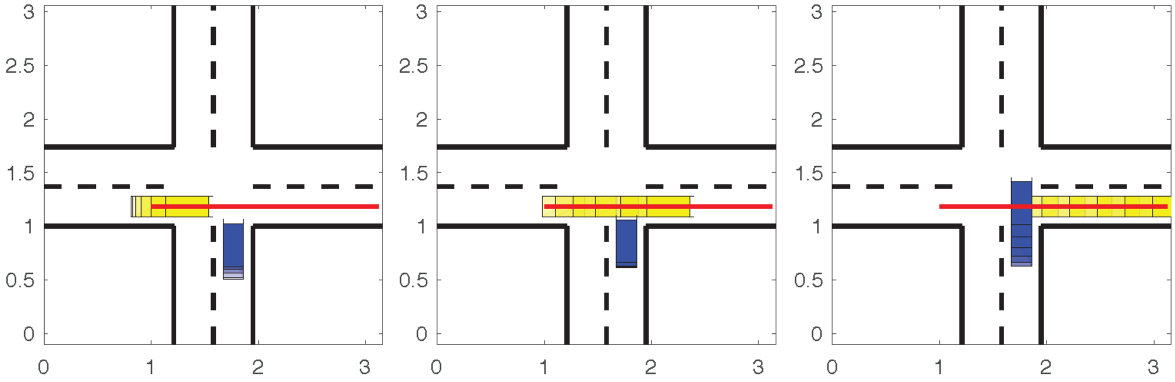

Figure 6.7: Negotiation with “nice” human driver in intersection scenario

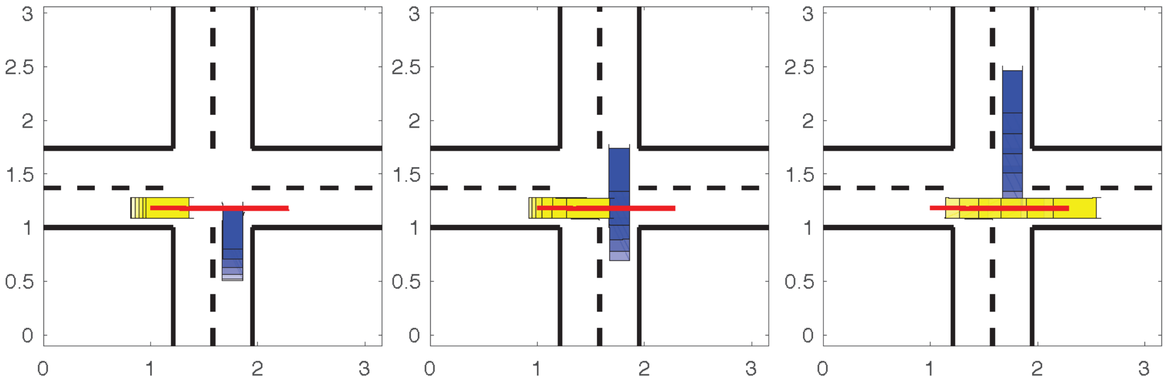

Figure 6.8: Negotiation with “tough” human driver in intersection scenario

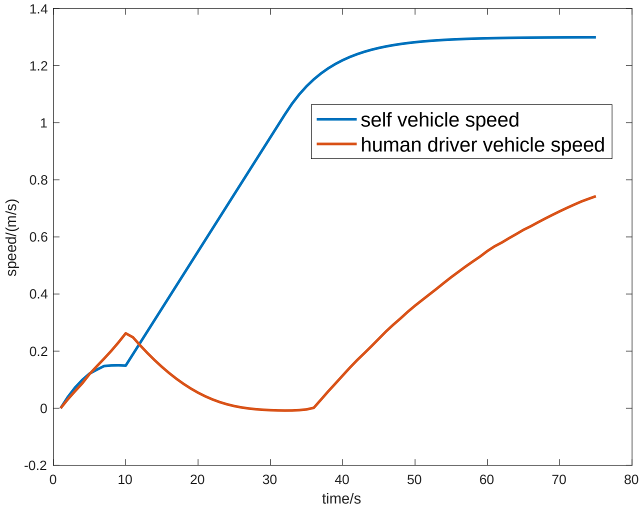

Figure 6.9: Intersection crossing speed profile (negotiation with “nice” human driver case)

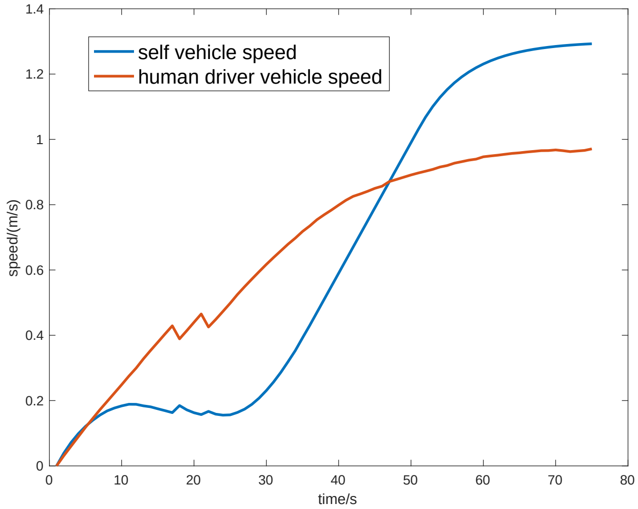

Figure 6.10: Intersection crossing speed profile (negotiation with “tough” human driver case)

vehicles, a concession weight cost function and a convex persuasion constraint were introduced to the model predictive control (MPC) optimization framework. With the concession weight cost function, the autonomous vehciel could adjust its own priority in the driving negotiation process according to human driver's reaction (defiend as acceptance probability of the cooperative proposal). The introduction of persuasion constraint enabled the autonomous vehicle to express its own intention and actively ask for right of the road. The proposed algorithm's performance was demonstrated by simulation results in several driving scenarios including lane changing, lane keeping and intersection crossing.

Chapter 7

Bayesian Persuasive Driving

In the previous chapter, the interactive driving problem is formulated as a negotiation based optimization which needs to solve for both vehicles. However, this will introduce extra computational load which is not desirable. In order to reduce the decision variables back to the ego vehicle only, the interacting surrounding vehicle's decision making process has to be modeled. In this chapter, a Bayesian persuasive driving algorithm is proposed, where the ego vehicle is the persuader (information sender) and the surrounding vehicle is the persuadee (information receiver). In the persuasion process, the ego vehicle aims at changing the surrounding vehicle's posterior belief of the world state by providing certain information via signaling in order to achieve a lower cost for both players. The information received by the surrounding vehicle and its belief of the world state are described by Gaussian distributions. Simulation results are provided to demonstrate the proposed algorithm's capability of handling interaction situations involving surrounding vehicles with different driving characteristics in several common traffic scenarios.

7.1 Introduction

Although autonomous vehicles have been spotted more and more frequently driving on the city roads, most of them are still not interacting with other road users like human drivers do. Instead, most autonomous vehicles are implementing a reactive behavior, which means that the trajectory predictions about surrounding vehicles are made first and the ego vehicle's driving actions are decided accordingly by applying obstacle avoidance algorithm. However, in this planning pattern, the interaction between vehicles is ignored since the interacting vehicle's future driving profile is assumed to be independent of the ego vehicle's behavior. Therefore, more efforts are needed to fulfill a real interactive, efficient and cooperative driving environment where robot cars and human drivers coexist.

Several approaches have been proposed for interactive driving in literatures, which can be categorized into two groups in general. The first category is multi-agent algorithm [10]-[19], where the assumption is made that all the vehicles involved in a driving scenario can be controlled. In a multi-agent system, all the vehicles optimize the same functional and each vehicle knows exactly what the others will be doing and what influence its behavior will cause. The major drawback of the multi-agent algorithm is that in the real world, not all the related vehicles can be controlled, especially at the current stage where autonomous vehicles only make a minority of all road users. Another disadvantage is its heavy dependency on vehicle-to-vehicle (V2V) communication which may not be reliable enough in real system.

The second category of algorithm is based on interactive prediction of surrounding vehicle's future behaviors. Several promising prediction approaches have been proposed in literatures including partially observable Markov decision process (POMDP), deep neural network, optimization-based method and so on. For example, a POMDP based decision making strategy was proposed in [22] for intersection scenario where interacting vehicles were assumed to pick one route from a predetermined route hypothesis set. However, this approach was confined to a specific driving situation due to the fixed route hypothesis setting. In [29], several deep neural network based motion models were evaluated for the highway entrance scenario, among which a model was selected for fast computation and relatively good performance. The drawback of this approach is that its performance heavily depends on feature selection which is not general for different driving scenarios. Optimization provides another direction of generating interactive prediction for surrounding vehicles. As an example, in [49], the human driver behavior is predicted by optimizing a reward function with pre-defined structure and parameters learned via inverse reinforcement learning (IRL) algorithm. Through including the learned human model reward function in robot vehicle trajecotry planning, the interaction effects can be handled. Compared to the POMDP and neural network based methods, this optimization-based approach is more computationally efficient. However, the learned reward function can only describe a particular type of driver, which is not a general solution to the interactive driving problem.

Another optimization-based interactive planning approach is to formulate the interactive driving problem as a Bayesian persuasion game, which is first proposed for economy application [24]. In the Bayesian persuasion game, there is one sender with information who attempts to persuade the receiver to change his/her action so that the welfare of both players can be improved. The basic assumptions include 1) the receiver's behavior is dependent on his/her belief of the world state and 2) both the players are rational Bayesian under which the interaction can be described as a Bayesian process. The persuasion process can be achieved by the sender via selecting certain information to convey to the receiver so that the receiver's posterior belief distribution of the world state can be properly manipulated.

In this chapter, the world state for the interactive driving environment is defined to be the ego vehicle's conservativeness perceived by the surrounding vehicle, the ego vehicle is defined as the information sender, whose information of driving intention can be reflected from his/her driving behavior and the surrounding interacting vehicle is the corresponding information receiver. With regard to the signal, several candidates are available including binary signal of yield or not yield, discrete signal of driving route selection and continuous signal of driving state. As a starting point, the ego vehicle's continuous driving state is selected as the signal since it carries the most driving information. By determining the optimal signal based on an optimization, the ego vehicle is able to achieve maximization/minimization of utility/cost expectation for both players. Meanwhile, the receiver will extract more information about the world state from the perceived signal and thus his/her posterior belief of the world state can be updated.

The remainder of the chapter is structured as follows. In Section 7.2, the mathematical formulation of the general Bayesian persuasion problem is introduced. In Section 7.3, the concrete Bayesian persuasion problem for interactive driving is formulated as an optimization with certain constraints incorporated including vehicle dynamics, safety and physical saturation limit. In addition, the integrals are calculated and the optimization problem is reformulated into a tractable form. In Section 7.4, the proposed algorithm's effectiveness is illustrated by simulations in several driving scenarios. Section 7.5 concludes the chapter.

7.2 General Bayesian Persuasion

In this section, the formulation of a general Bayesian persuasion game is introduced and the related preliminary notations are given.

A general and intuitive definition of the persuasion problem is to exploit some information advantage to influence the opponent's action or intention. In fact, the persuasion behavior is ubiquitous in everyday life with applications in a great deal of areas including economy, psycology, decision making theory and so on. Basically, in almost any interacting process, there always exists a persuasion scheme which is advantageous to some or all players.

Among various persuasion models, the Bayesian persuasion model [24] first proposed for economy application stands out as the most popular and fundamental one. In a Bayesian persuasion game, there are two players: the player sending information is called sender and the other one receiving information is called receiver. The sender aims to change the receiver's action so that there is a higher probability that a situation more beneficial to both parties can be achieved. The receiver's task is to pick an action based on the information extracted from the sender's signal. The reward/cost function of the game depends on both players' actions meaning that neither player can determine the game's result by himself. This characteristic leads to the fact that both players cannot exactly know what the reward or cost will be until both of their actions have been unveiled.

In order to formulate the persuasion game mathematically, some notations are introduced first. The action of the receiver is denoted as $a \in A$, where A is the receiver's action space. The world state is represented by $\omega \in \Omega$, where Ω is the world state space. The realization space of the sender's information signal is denoted as S and the corresponding signal realization is s. However, the information carried by the signal may not be fully comprehended by the receiver. In the receiver's point of view, the signal should be described by a probability distribution which is called signal belief distribution denoted by $\pi(s)$. Hence, $\pi(s)$ represents the distribution of the information signal perceived by the receiver. Due to the similar reason, another world state belief distribution $\mu(\omega|s)$ is introduced to describe how the world state ω is influenced by the signal in the receiver's mind. For the game's objective function, maximizing a reward function and minimizing a cost function are equivalent. Thus, without loss of generality, we choose to define a cost function $c(\omega, a, s)$, which is dependent on the world state, receiver's action and the sender's signal.

With the notations defined above, the Bayesian persuasion problem can be formulated as:

$$\min_{\pi(s)} \quad E_{\pi(s)}\hat{c}(\mu(\omega|s)), \tag{7.1}$$

where

$$\hat{c}(\mu(\omega|s)) = \min_{a} E_{\mu(\omega|s)} c(\omega, a, s)$$

represents the expected cost at a specific signal realization s when the receiver holds the belief distribution $\mu(\omega|s)$. The solution to the optimization (7.1) is defined as the optimal signal and the corresponding achieved minimum value is called the value of the optimal signal. In the Bayesian persuasion process described by (7.1), according to the rationality assumption, the receiver will decide his action by optimizing the objective function expectation given his belief of the world state influenced by the sender's signal. Moreover, since the sender is aware that the receiver's rational, he can then determine the optimal signal to send based on the receiver's strategy.

In summary, the optimization problem in (7.1) illustrates the idea that the sender's purpose is to minimize both players' cost by manipulating the receiver's posterior belief distribution $\mu(\omega|s)$ via conveying information via signaling perceived by the receiver as a probability distribution $\pi(s)$.

7.3 Bayesian Persuasive Driving

In this section, the Bayesian persuasion framework is applied to the interacting driving problem with Gaussian distribution assumption. The concrete definition of variables are given in the autonomous driving context first and a mathematical approximation is then applied to the resulting driving persuasion problem to make it tractable.

Bayesian Persuasion in Autonomous Driving Context

In order to formulate a Bayesian persuasion game in the autonomous driving background, the players along with their possible actions and signals need to be clearly defined first.

Intuitively, the surrounding interacting vehicle is selected as the information receiver and his driving state

$$\mathbf{x}_t^s = [\ x_t^s \quad y_t^s \quad \theta_t^s \quad v_t^s\ ]^T$$

is defined as his action a_t, where x_t^s, y_t^s, θ_t^s and v_t^s denote the surrounding vehicle's $x-y$ positions, yaw angle and speed in the lane-based coordinate frame at time instant t respectively. The information sender role is then naturally assigned to the ego vehicle. With regard to the signaling content, there are quite a few options including intention indicator for yielding or not yielding, route selection preference and the driving state itself. In this chapter, the ego vehicle's driving state

$$\mathbf{x}_t^e = [\ x_t^e \quad y_t^e \quad \theta_t^e \quad v_t^e\ ]^T$$

is chosen as the signal realization s_t since it includes more intention/driving behavior information and it is directly perceivable for the surrounding vehicle. Similar with the notation for the surrounding vehicle, x_t^e, y_t^e, θ_t^e and v_t^e denote the ego vehicle's $x-y$ positions, yaw angle and speed in the lane-based coordinate frame at time instant t respectively.

The last definition involved in the Bayesian persuaison is the world state variable ω. The characteristic of the Bayesian persuasion game imposes two requirements on the state ω, which are 1) it determines the cost for both players along with the receiver's action and the sender's signal, 2) it cannot be directly influenced by the receiver's action. In order to satisfy the two mentioned properties, the state is defined as the ego vehicle's conservativeness perceived by the surrounding vehicle. Basically, the state ω denotes the receiver (surrounding vehicle)'s impression of the sender (ego vehicle), whether aggressive, conservative or in between. The mathematical formulation of ω_t is as:

$$\omega_t = \|I'(\mathbf{x}_t^{s,p} - \mathbf{x}_t^{e,p})\|_2, \tag{7.2}$$

where $\mathbf{x}_t^{s,p} = [\ x_t^{s,p} \quad y_t^{s,p} \quad \theta_t^{s,p} \quad v_t^{s,p}\ ]^T$ denotes the surrounding vehicle's predicted driving state at time step t and $\mathbf{x}_t^{e,p}$ is the ego vehicle's predicted driving state made by the surrounding vehicle defined similarly with $\mathbf{x}_t^{s,p}$. The definition of I' is as

$$I' = \begin{bmatrix} \frac{1}{a_s} & 0 & 0 & 0 \\ 0 & \frac{1}{b_s} & 0 & 0 \end{bmatrix},$$

where a_s, b_s are semi-major axis and semi-minor axis of the ellipse representing the surrounding vehicle respectively. The definition (7.2) shows that ω is a scalar and the smaller it is, the less conservative (more aggressive) the ego vehicle appears to the surrounding vehicle. For instance, in the extreme scenario where the distance d_t is almost 0 and ω equals to 0 which means the ego vehicle totally does not care about the surrounding vehicle so that its behavior becomes significantly influential to the surrounding vehicle. An opposite case happens when d_t is approaching infinity and ω also goes to infinity, the ego vehicle then cannot bring any influence to the surrounding vehicle and the surrounding vehicle thus has no incentive to consider what the ego vehicle will do in the future. Of course, these scenarios are impossible in the practical driving system. However, they effectively demonstrate the essence of ω_t.

In summary, in the Bayesian persuasive driving process, the ego vehicle aims at finding an optimal signal determined by its driving behavior. The general Bayesian persuasion optimization (7.1) can be reformulated as

$$\min_{\pi(\mathbf{x}_t^e),\mathbf{u}_t^e} \quad \int_{\mathbf{x}_t^e} \pi(\mathbf{x}_t^e) \min_{\mathbf{x}_t^s} \int_{\omega_t} \mu(\omega_t|\mathbf{x}_t^e) c(\omega_t, \mathbf{x}_t^s, \mathbf{x}_t^e, \mathbf{u}_t^e) \tag{7.3}$$

for the interactive driving scenario with the expectation term expanded, where

$$\mathbf{u}_t^e = [\ a_t^e \quad \delta_t^e\ ]^T$$

denotes the ego vehicle's control input at time instant t including acceleration a_t^e and steering angle δ_t^e. In order to avoid shortsighted non-optimal behavior, a receding time horizon is introduced:

$$\begin{aligned}\min_{\pi(\mathbf{x}^e_{t|t_0}),\mathbf{u}^e_{t|t_0}} \quad & \sum_{t=t_0}^{t_0+N}\int_{\mathbf{x}^e_{t|t_0}} \pi(\mathbf{x}^e_{t|t_0}) \times \\ & \min_{\mathbf{x}^s_{t|t_0}} \int_{\omega_{t|t_0}} \mu(\omega_{t|t_0}|\mathbf{x}^e_{t|t_0}) c(\omega_{t|t_0},\mathbf{x}^s_{t|t_0},\mathbf{x}^e_{t|t_0},\mathbf{u}^e_{t|t_0}),\end{aligned} \tag{7.4}$$

where N is the optimization horizon length, t_0 is the current time instant, $(\bullet)_{t|t_0}$ denotes prediction of variable for t made at t_0.

Gaussian Assumption

The optimization (7.3) is intractable since the decision variable is a probability distribution in continuous space. In order to make (7.3) solvable, the Gaussian assumption is made so that each probability distribution can be described by a mean and a variance in an exponential form. Therefore, the decision variable of (7.3) is reduced from a complicated probability distribution to a vector-valued mean and a covairance matrix. Moreover, the exponential form of Gaussian distribution also facilitates the next integral approximation step. With the Gaussian assumption, the probability distributions π and μ in the original problem (7.4) can be explicitly written as

$$\begin{aligned}&\pi : \mathbf{x}^e_{t|t_0} \sim \mathcal{N}(\hat{\mathbf{x}}^e_{t|t_0}, \Sigma_{\mathbf{x}^e_{t|t_0}}),\\ &\mu : \omega_{t|t_0}|\mathbf{x}^e_{t|t_0} \sim \mathcal{N}(\hat{\omega}_{t|t_0}|\mathbf{x}^e_{t|t_0}, \Sigma_{\omega_t|\mathbf{x}^e_{t|t_0}}),\end{aligned} \tag{7.5}$$

where $x \sim \mathcal{N}(\hat{x},\Sigma)$ means that random variable x has the Gaussian distribution with mean of $\hat{x}$ and covariance of Σ.

Then the Bayesian persuasion game can be reorganized as:

$$\begin{aligned}\min_{\hat{\mathbf{x}}^e_{t|t_0},\mathbf{u}^e_{t|t_0}} \quad & \sum_{t=t_0}^{t_0+N}\int_{\mathbf{x}^e_{t|t_0}} G_{\mathbf{x}^e_{t|t_0}}(\hat{\mathbf{x}}^e_{t|t_0},\Sigma_{\mathbf{x}^e_{t|t_0}}) \times \\ & \min_{\mathbf{x}^s_{t|t_0}} \int_{\omega_{t|t_0}} \Big[G_{\omega_{t|t_0}}(\hat{\omega}_{t|t_0}|\mathbf{x}^e_{t|t_0},\Sigma_{\omega_{t|t_0}|\mathbf{x}^e_{t|t_0}}) \times \\ & \qquad c(\omega_{t|t_0},\mathbf{x}^s_{t|t_0},\mathbf{x}^e_{t|t_0},\mathbf{u}^e_{t|t_0})\Big]\end{aligned} \tag{7.6}$$

where the expectations of ego vehicle driving states $\hat{\mathbf{x}}^e_{t|t_0}(t = t_0,\cdots,t_0+N)$ and control inputs $\mathbf{u}^e_{t|t_0}(t = t_0,\cdots,t_0+N)$ are new decision variables,

$$G_x(\hat{x},\Sigma) = \frac{exp(-\frac{1}{2}(x-\hat{x})^T\Sigma^{-1}(x-\hat{x}))}{\sqrt{(2\pi)^k|\Sigma|}},$$

denotes the density function of Gaussian distribution $\mathscr{N}(\hat{x},\Sigma)$ and k is the dimension of x. $\hat{\omega}_{t|t_0}|\mathbf{x}^e_{t|t_0}$ follows the same definition of ω as in (7.2):

$$\hat{\omega}_{t|t_0}|\mathbf{x}^e_{t|t_0} = \|I'(\mathbf{x}^{s,p}_t - \mathbf{x}^e_{t|t_0})\|_2, \tag{7.7}$$

where the ego vehicle's driving state prediction $\mathbf{x}^{e,p}_t$ is replaced by $\mathbf{x}^e_{t|t_0}$. According to the definition (7.7), the surrounding vehicle's expected impression of the ego vehicle given his driving behavior is dependent on how the ego vehicle will influence its original driving plan.

Cost Function

The cost function c in (7.6) is defined in exponential form as

$$\begin{aligned} c(\omega_{t|t_0}, \mathbf{x}^s_{t|t_0}, \mathbf{x}^e_{t|t_0}, \mathbf{u}^e_{t|t_0}) \;\; = \;\; & exp\Big((1+\omega_{t|t_0})\|\mathbf{x}^s_{t|t_0} - \mathbf{x}^{s,p}_{t|t_0}\|^2_{W_1} \\ & +\|\mathbf{x}^e_{t|t_0} - \mathbf{x}^e_g\|^2_{W_2} + \|\Delta\mathbf{u}^{e,T}_{t|t_0}\|_{W_3} \\ & -\|\mathbf{x}^s_{t|t_0} - \mathbf{x}^e_{t|t_0}\|^2_{W_4}\Big), \end{aligned} \tag{7.8}$$

where $\mathbf{x}^e_g$ denotes the ego vehicle's desired goal state, $\Delta\mathbf{u}^e_{t|t_0} = \mathbf{u}^e_{t|t_0} - \mathbf{u}^e_{t-1|t_0}$ represents the change of ego vehicle's control input and W_1, W_2, W_3 and W_4 are positive definite penalty matrices. It is required that $W_1 - W_4 \succeq 0$ in order to guarantee the existence of a minimum for the cost function c with regard to $\mathbf{x}^s_{t|t_0}$. In the rest of the paer, W_1 and W_4 are set to be equal to $w_1 I$ and $w_4 I$ respectively, where I denotes the identity matrix, w_1 and w_4 are scalars.

The first term in the cost function (7.8) represents the surrounding vehicle's preference of tracking his original driving plan, the second and third terms are driving the ego vehicle to his goal and penalizing the input change in order to achieve comfortable driving experience and the last term represents the surrounding vehicle's aversion of risk, i.e., the preference to keep a certain distance from the ego vehicle. According to the definition (7.8), when the perceived conservativeness of the ego vehicle $\omega_{t|t_0}$ is lower, the surrounding vehicle will be inclined to focus more on the safety instead of sticking to his original plan. Otherwise, when $\omega_{t|t_0}$ is higher, meaning that the surrounding vehicle is more likely to treat the ego vehicle as a conservative agent, it will be intuitive for him to pursue a more selfish behavior.

The cost function c in (7.8) shows that the surrounding vehicle's action is dependent on two factors, i.e., the world state of the Bayesian game $\omega_{t|t_0}$ and the penalty matrix W_1. The penalty matrix W_1 represents the interacting vehicle's driving characteristics, which can only be recognized but not controlled.

Note that although the cost function (7.8) is intended for two vehicle interaction scenario, the framework can be extended to multiple vehicle interaction case by including more surrounding vehicles in the cost function definition.

Constraints

In the interactive driving application, besides the Bayesian cost function in (7.6), certain constraints need to be handled including model dynamics, control input saturation and safety constraint.

Vehicle Dynamics

In this chapter, the bicycle model [46] is adopted to describe the vehicle dynamics as follows:

$$\begin{aligned}
&x^e_{t+1|t_0}{=}x^e_{t|t_0}{+}T_s v^e_{t|t_0}\cos\left(\theta^e_{t|t_0}{+}\tan^{-1}(\frac{L_r}{L}\tan\delta^e_{t|t_0})\right)\\
&y^e_{t+1|t_0}{=}y^e_{t|t_0}{+}T_s v^e_{t|t_0}\sin\left(\theta^e_{t|t_0}{+}\tan^{-1}(\frac{L_r}{L}\tan\delta^e_{t|t_0})\right)\\
&\theta^e_{t+1|t_0}{=}\theta^e_{t|t_0}{+}T_s v^e_{t|t_0}\frac{\tan\delta^e_t}{L}\cos\left(\tan^{-1}(\frac{L_r}{L}\tan\delta^e_{t|t_0})\right)\\
&v^e_{t+1|t_0}{=}v^e_{t|t_0}{+}T_s a^e_{t|t_0},
\end{aligned}\tag{7.9}$$

where T_s is the sampling time, t is the time index, L_r, L_f and $L{=}L_r{+}L_f$ denote the dimension parameters of the vehicle which are the vehicle's rear, front and full length respectively.

The model equations (7.9) can be summarized as

$$\mathbf{x}^e_{t+1|t_0} = f(\mathbf{x}^e_{t|t_0}, \mathbf{u}^e_{t|t_0}).\tag{7.10}$$

Safety Constraint

Another critical constraint for autonomous driving is the guarantee of safety. For the static obstacles including parking vehicles and lane boundaries, the following constraint is defined:

$$y_{min} \le y^{e,i}_{t|t_0} \le y_{max}, i \in \{1,2,3,4\},\tag{7.11}$$

where $y^{e,i}_{t|t_0}$ denotes the y coordinate of the vehicle's i-th corner at time step t, y_{min} and y_{max} represent the lateral position's lower and upper limit respectively. With regard to the moving surrounding vehicle obstacles, the safety constraint is defined as:

$$\|I'(\mathbf{x}^{s,p}_t - \mathbf{x}^e_{t|t_0})\|_2 \ge 1,\tag{7.12}$$

where the surrounding vehicle is described by the same ellipse as in (7.2). In addition, another constraint is imposed on the vehicle speed which is

$$v_{min} \le v^e_{t|t_0} \le v_{max},\tag{7.13}$$

where v_{min} and v_{max} are minimum and maximum speed respectively.

Saturation Constraint

In addition to the vehicle modeling and safety constraint, the system also needs to be consistent with the physical control saturation constraint described by

$$\underline{\mathbf{u}}^e \leq \mathbf{u}^e_{t|t_0} \leq \overline{\mathbf{u}}^e, \tag{7.14}$$

where $\underline{\mathbf{u}}^e$, $\overline{\mathbf{u}}^e$ represent the lower and upper control saturation bound respectively.

Integral Calculation

Currently, the Bayesian persuasion is mainly applied in the economics community. The main factor limiting its popularity in other areas is that calculation of expectation for continuous distribution requires computation of integrals as illustrated by (7.6). Although Gaussian assumption grants the reduction of decision variable from a distribution to a vector, calculation of integrals is still challenging, especially for the autonomous driving problem with high dimension involved. In this subsection, through introducing several approximations, the original cost fucntion (7.4) is reformulated into a tractable form. By an abuse of notation, the subscript $t|t_0$ is replaced by t in this subsection.

First consider the integral with regard to ω_t:

$$\int_{\omega_t} G_{\omega_t}(\hat{\omega}_t|\mathbf{x}^e_t, \Sigma_{\omega_t|\mathbf{x}^e_t}) c(\omega_t, \mathbf{x}^s_t, \mathbf{x}^e_t, \mathbf{u}^e_t). \tag{7.15}$$

As ω_t is a scalar variable, we can directly calculate the result of (7.15) as

$$\begin{aligned} & \int_{\omega_t} G_{\omega_t}(\hat{\omega}_t|\mathbf{x}^e_t, \Sigma_{\omega_t|\mathbf{x}^e_t}) c(\omega_t, \mathbf{x}^s_t, \mathbf{x}^e_t, \mathbf{u}^e_t) \\ = \; & \varepsilon exp\left(\frac{1}{2}\Sigma_{\omega_t|\mathbf{x}^e_t}[(\mathbf{x}^s_t - \mathbf{x}^{s,p}_t)^T W_1 (\mathbf{x}^s_t - \mathbf{x}^{s,p}_t)]^2\right) \times \\ & c(\hat{\omega}_t|\mathbf{x}^e_t, \mathbf{x}^s_t, \mathbf{x}^e_t, \mathbf{u}^e_t), \end{aligned} \tag{7.16}$$

where

$$\varepsilon = \Psi\left(\frac{\hat{\omega}_t|\mathbf{x}^e_t + \Sigma_{\omega_t|\mathbf{x}^e_t}(\mathbf{x}^s_t - \mathbf{x}^{s,p}_t)^T W_1 (\mathbf{x}^s_t - \mathbf{x}^{s,p}_t)}{\Sigma^{\frac{1}{2}}_{\omega_t|\mathbf{x}^e_t}}\right),$$

$$\Psi(x) = \frac{1}{2}(1 + erf(x/\sqrt{2}))$$

is the cumulative distribution function of a standard Gaussian distribution and $erf(\bullet)$ is the error function. According to the safety constraint (7.12), $\hat{\omega}_t|x^e_t \geq 1$ always holds. Hence when the variance $\Sigma_{\omega_t|\mathbf{x}^e_t}$ is chosen to be less than 0.2, we have

$$1 \geq \varepsilon \geq \Psi(\sqrt{5}) = \frac{1}{2}(1 + erf(\sqrt{2.5})) = 0.9873, \tag{7.17}$$

resulting in $\varepsilon \approx 1$ due to $erf(\bullet)$'s S-shape property.

Then, the solution to the sub-problem

$$\min_{\mathbf{x}_t^s} \int_{\omega_t} G_{\omega_t}(\hat{\omega}_t|\mathbf{x}_t^e, \Sigma_{\omega_t|\mathbf{x}_t^e}) c(\omega_t, \mathbf{x}_t^s, \mathbf{x}_t^e, \mathbf{u}_t^e)$$

can be obtained as

$$\begin{aligned} \mathbf{x}_t^{s*} \approx & \left((1+\Sigma_{\omega_t|\mathbf{x}_t^e}\gamma_t + \hat{\omega}|\mathbf{x}_t^e)W_1 - W_4\right)^{-1} * \\ & \left((1+\Sigma_{\omega_t|\mathbf{x}_t^e}\gamma_t + \hat{\omega}|\mathbf{x}_t^e)W_1\mathbf{x}_t^{s,p} - W_4\mathbf{x}_t^e\right) \end{aligned} \tag{7.18}$$

via setting the derivative equal to zero, where

$$\gamma_t = (\mathbf{x}_{t-1}^s - \mathbf{x}_{t-1}^{s,p})^T W_1 (\mathbf{x}_{t-1}^s - \mathbf{x}_{t-1}^{s,p}),$$

$\mathbf{x}_{t-1}^{s,p}$ is the previous prediction and $\mathbf{x}_{t-1}^s$ denotes the surrounding vehicle's previous state.

Substituting (7.18) into (7.16) obtains

$$\begin{aligned} & \min_{\mathbf{x}_t^s} \int_{\omega_t} G_{\omega_t}(\hat{\omega}_t|\mathbf{x}_t^e, \Sigma_{\omega_t|\mathbf{x}_t^e}) c(\omega_t, \mathbf{x}_t^s, \mathbf{x}_t^e) \\ \approx \; & exp\{(\mathbf{x}_t^e - \mathbf{x}_g^e)^T W_2 (\mathbf{x}_t^e - \mathbf{x}_g^e) + \mathbf{u}_t^{e,T} W_3 \mathbf{u}_t^e + k_1 \hat{\omega}_t|\mathbf{x}_t^e + k_2 \\ & + \frac{k_3}{[w_1\hat{\omega}_t|\mathbf{x}_t^e + ((1+\Sigma_{\omega_t|\mathbf{x}_t^e}\gamma_t)w_1 - w_4)]} \\ & + \frac{k_4}{[w_1\hat{\omega}_t|\mathbf{x}_t^e + ((1+\Sigma_{\omega_t|\mathbf{x}_t^e}\gamma_t)w_1 - w_4)]^2}\}, \end{aligned} \tag{7.19}$$

where k_1, k_2, k_3 and k_4 are constant scalars dependent on w_1, w_4, $\Sigma_{\omega_t}|\mathbf{x}_t^e$ and γ_t. Furthermore, it can be guaranteed that k_3 is always negative and k_4 is always positive so that there exists a upper bound $\bar{k}$ for the last two terms in (7.19) as $\hat{\omega}_t|\mathbf{x}_t^e \geq 0$.

Thus the original optimization (7.6) is changed to

$$\begin{aligned} \min_{\hat{\mathbf{x}}_t^e, \mathbf{u}_t^e} \int_{\mathbf{x}_t^e} & G_{\mathbf{x}_t^e}(\hat{\mathbf{x}}_t^e, \Sigma_{\mathbf{x}_t^e}) exp\{(\mathbf{x}_t^e - \mathbf{x}_g^e)^T W_2 (\mathbf{x}_t^e - \mathbf{x}_g^e) \\ & + \mathbf{u}_t^{e,T} W_3 \mathbf{u}_t^e + k_1 \hat{\omega}_t|\mathbf{x}_t^e\}, \end{aligned} \tag{7.20}$$

where the constant terms k_2 and $\bar{k}$ are omitted.

The integral term in (7.20) can be compactly written as

$$\int_{\mathbf{x}_t^e} exp(C(\mathbf{x}_t^e)),$$

where $C(\mathbf{x}_t^e)$ is in quadratic form. Since the variable $\mathbf{x}_t^e$ is a vector, a mathematical approximation technique based on Lapace's method is utilized to eliminate the high-dimensional integral [12]. Via taking second order Taylor seires expansion around $\mathbf{x}_t^{e*}{:=}arg\min_{\mathbf{x}_t^e} C(\mathbf{x}_t^e)$ where $\nabla C(\mathbf{x}_t^{e*})$ equals to 0, the following equation can be obtained:

$$\begin{aligned} C(\mathbf{x}_t^e) &\approx C(\mathbf{x}_t^{e*}) + \nabla C(\mathbf{x}_t^{e*})^T(\mathbf{x}_t^e - \mathbf{x}_t^{e*}) \\ &\quad + \frac{1}{2}(\mathbf{x}_t^e - \mathbf{x}_t^{e*})^T \nabla^2 C(\mathbf{x}_t^{e*})(\mathbf{x}_t^e - \mathbf{x}_t^{e*}) \\ &= C(\mathbf{x}_t^{e*}) + \frac{1}{2}(\mathbf{x}_t^e - \mathbf{x}_t^{e*})^T \nabla^2 C(\mathbf{x}_t^{e*})(\mathbf{x}_t^e - \mathbf{x}_t^{e*}). \end{aligned} \tag{7.21}$$

Then the integral term $\int_{\mathbf{x}_t^e} exp(C(\mathbf{x}_t^e))$ can be approximated as

$$\int_{\mathbf{x}_t^e} exp(C(\mathbf{x}_t^e)) \approx k_e exp(C(\mathbf{x}_t^{e*})), \tag{7.22}$$

where k_e is a constant determined by the Hessian $\nabla^2 C(\mathbf{x}_t^{e*})$. Then the cost function in (7.20) can be reformulated as

$$\begin{aligned} \min_{\hat{\mathbf{x}}_t^e, \mathbf{u}_t^e} \ J_t &= -\frac{1}{2}(\mathbf{x}_t^{e*} - \hat{\mathbf{x}}_t^e)^T \Sigma_{\mathbf{x}_t^e}^{-1}(\mathbf{x}_t^{e*} - \hat{\mathbf{x}}_t^e) + \mathbf{u}_t^{e,T} W_3 \mathbf{u}_t^e \\ &\quad + (\mathbf{x}_t^{e*} - \mathbf{x}_g^e)^T W_2 (\mathbf{x}_t^{e*} - \mathbf{x}_g^e) + k_1 \hat{\omega}_t | \mathbf{x}_t^{e*}. \end{aligned} \tag{7.23}$$

Note that $\mathbf{x}_t^{e*}$ in (7.23) is a linear combination of $\hat{\mathbf{x}}_t^e$, $\mathbf{x}_g^e$ and $\mathbf{x}_t^{s,p}$, thus the cost function in (7.23) is a quadratic function with regard to $\hat{\mathbf{x}}_t^e$.

Remark 1: Note that the accurate distribution of $\omega_t|\mathbf{x}_t^e$ is supposed to be truncated Gaussian instead of standard Gaussian as ω_t is always positive due to defintion in (7.2). The strict formula of $\omega_t|\mathbf{x}_t^e$'s density function is

$$f(\omega|\mathbf{x}_t^e) = \frac{G_{\omega_t}(\hat{\omega}_t|\mathbf{x}_t^e, \Sigma_{\omega_t})}{\Sigma_{\omega_t|\mathbf{x}_t^e}^{\frac{1}{2}}\left(1 - \Psi\left(-\frac{\hat{\omega}_t|\mathbf{x}_t^e}{\Sigma_{\omega_t|\mathbf{x}_t^e}^{\frac{1}{2}}}\right)\right)}. \tag{7.24}$$

According to the safety consrtaint (7.12) and utilizing the same approximation as for ε in (7.17), the denominator of (7.24) can be treated as a constant and hence the usage of Gaussian distribution in (7.6) is validated.

Bayesian Persuasive Optimization

With the integral calculation result from the previous subsection, the solvable form of the original Bayesian persuasive driving optimization (7.4) can be summarized as

$$\min_{\hat{\mathbf{x}}^e_{t|t_0},\mathbf{u}^e_{t|t_0}} \quad \textstyle\sum_{t=t_0}^{t_0+N-1} J_t \tag{7.25}$$

$$\begin{aligned} s.t. \quad & \hat{\mathbf{x}}^e_{t+1|t_0} = f(\hat{\mathbf{x}}^e_{t|t_0},\mathbf{u}^e_{t|t_0}), \\ & \|I'(\hat{\mathbf{x}}^e_{t|t_0} - \mathbf{x}^{s,p}_t)\|_2 \geq 1, \\ & \underline{\mathbf{u}}^e \leq \mathbf{u}^e_{t|t_0} \leq \overline{\mathbf{u}}^e, \\ & \hat{\mathbf{x}}^e_{t_0|t_0} = \mathbf{x}^e_{t_0}, v_{min} \leq \hat{v}^e_{t|t_0} \leq v_{max}, \\ & y_{min} \leq \hat{y}^{e,i}_{t|t_0} \leq y_{max}, i \in \{1,2,3,4\}, \\ & t = t_0, \cdots, t_0+N-1, \end{aligned} \tag{7.26}$$

where $\mathbf{x}^e_{t_0}$ is the ego vehicle's current driving state.

Remark 2: After obtaining the solution to the optimization (7.25) and observing the surrounding vehicle's behavior, more information about the interacting vehicle's characteristics will be revealed so that the covariance and penalty matrices in the cost function should be updated accordingly. With regard to $\Sigma_{\mathbf{x}^e_t}$, it is updated based on the surrounding vehicle's confident level of the ego vehicle's behavior, which can be approximated by

$$\alpha = \sum_{t=t_0}^{t_0+N-1} w^\alpha_t \|\mathbf{x}^e_{t|t_0} - \mathbf{x}^e_{t|t_0-1}\|_2, \tag{7.27}$$

where w^α_t's are weighting factors satisfying $\sum_t w^\alpha_t = 1$. According to the definition, smaller α indicates that the ego vehicle's behavior is more consistent with its previous driving plan and thus the surrounding vehicle will be more confident about the information he extracts from the ego vehicle's behavior. Therefore, the covariance matrix of probability distribution $\Sigma_{\mathbf{x}^r_t}$ should be proportional to α.

As shown in the cost function (7.8), the penalty matrix W_1 which reflects the surrounding vehicle's driving characteristics is another influential factor of the algorithm. Similar with the update strategy for $\Sigma_{\mathbf{x}^e_t}$, the update of W_1 is based on the surrounding vehicle's confidence in his own driving plan, which can be inferred from the following parameter:

$$\beta = \sum_{t=t_0}^{t_0+N-1} w^\beta_t \|\mathbf{x}^{s,p}_{t|t_0} - \mathbf{x}^{s,p}_{t|t_0-1}\|_2, \tag{7.28}$$

where w^β_t's are weighting factors for β satisfying $\sum_t w^\beta_t = 1$. Intuitively, a smaller β represents a smaller change of the surrounding vehicle's driving plan, reflecting his more self-centric driving characteristics resulting in a larger W_1.

7.4 Simulation Results

In this section, the proposed algorithm's performance is demonstrated by simulations implemented for several driving scenarios including lane changing, lane keeping and intersection crossing. The setting details for the simulation are the same as in Table 6.1 and Table 6.2. In the simulation, the surrounding vehicles are assumed to follow a model predictive control (MPC) strategy with different safety weights reflecting various driving characteristics, i.e., more cautious/conservative drivers are associated with higher safety weights and more aggressive drivers are represented by lower safety weights. Another assumption is about perfect prediction, which means that the ego vehicle knows exactly what the surrounding vehicle's driving plan is. Although this is a very strong assumption, the algorithm is still necessary as its point is to persuade the surrounding vehicle to change its original plan instead of just identifying it.

Lane Changing Scenario

First consider the scenario where the ego vehicle attempts to change to the neighboring lane with two surrounding vehicles running in it. The front surrounding vehicle is assumed to be always aggressive (safety weight equals to zero) so that it will never yield the ego vehicle and the rear vehicle's safety weight is adjusted to represent different kinds of driver, where large safety weight means nice driver and small or zero safety weight is utilized for iron nerved aggressive driver. The interacting case with "nice" driver is shown in Fig. 7.1, where the ego vehicle is represented by the yellow rectangle, the front and rear vehicle are plotted as blue and green rectangle respectively. The solid red line represents the ego vehicle's driving trajectory. In this case, the nice driver decided to decelerate and yield the ego vehicle. The ego vehicle thus took the chance and finished the lane changing task smoothly.

Figure 7.2 shows another case with "tough" driver who chooses to accelerate and ignore the ego vehicle's lane changing need. In this case, as the rear vehicle refused to yield, the ego vehicle waited until the rear surrounding vehicle passed and completed lane changing later.

Lane Keeping Scenario

In the lane keeping scenario, the ego vehicle needs to make decision between to yield or not to yield when the surrounding vehicle seeks to merge in. The simulation results Fig. 7.3 and Fig. 7.4 illustrate the ego vehicle's intelligent driving behavior when interacting with nice and tough driver respectively. In the case of a nice driver, the ego vehicle decides to ignore its merging request and accelerate to show its intention as shown in Fig. 7.3. On the other hand, when interacting with a more aggressive driver, the ego vehicle expresses its intention of yielding by decelerating as shown in Fig. 7.4. The speed profile for these two driving situations are shown in Fig. 7.5 and Fig. 7.6 respectively.

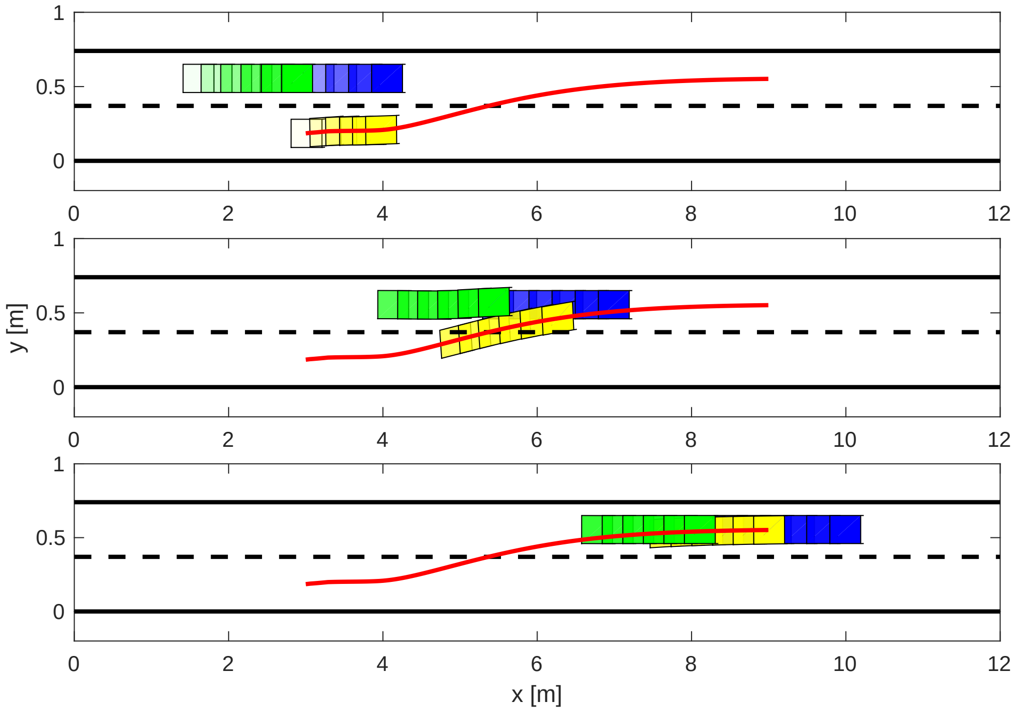

Figure 7.1: Bayesian persuasion with "nice" surrounding vehicle in lane changing scenario

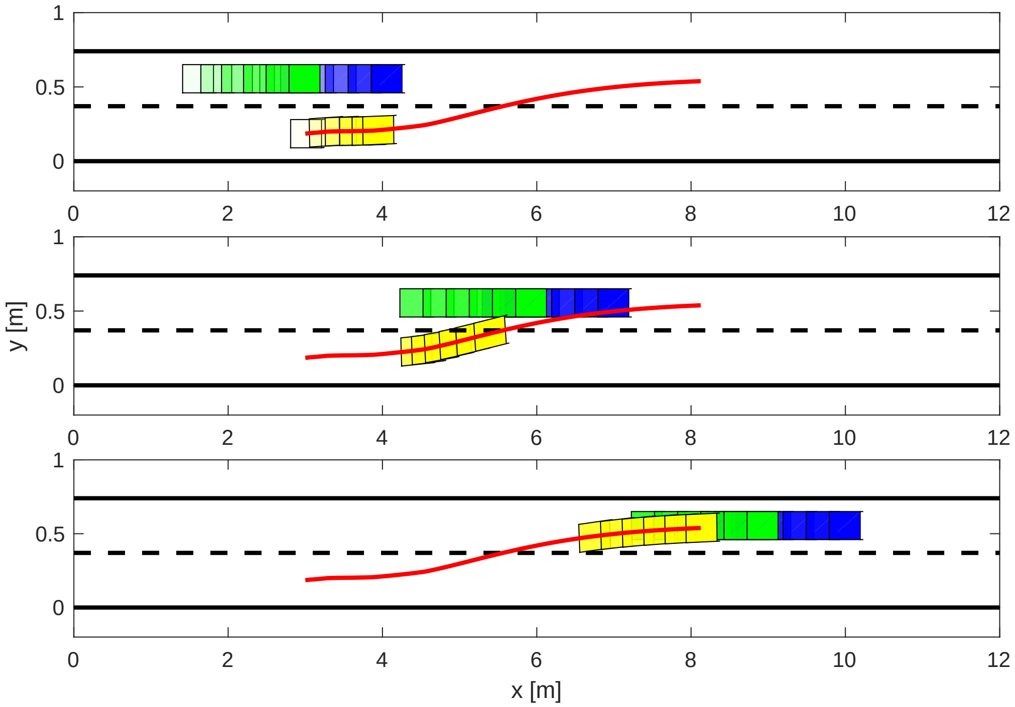

Figure 7.2: Bayesian persuasion with "tough" surrounding vehicle in lane changing scenario

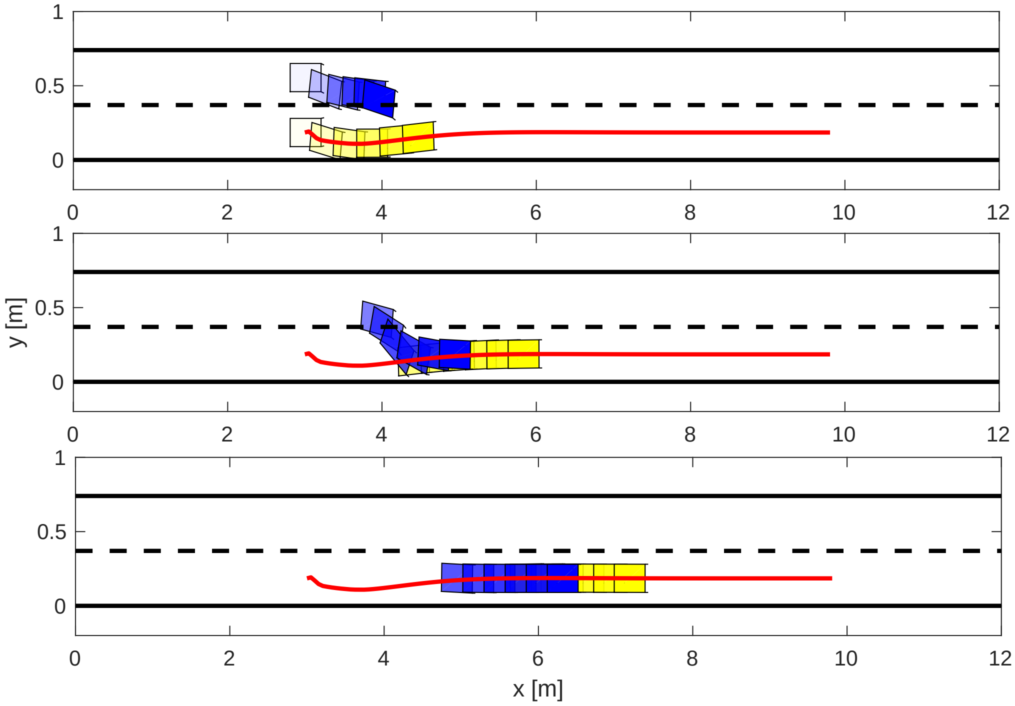

Figure 7.3: Bayesian persuasion with "nice" surrounding vehicle in lane keeping scenario

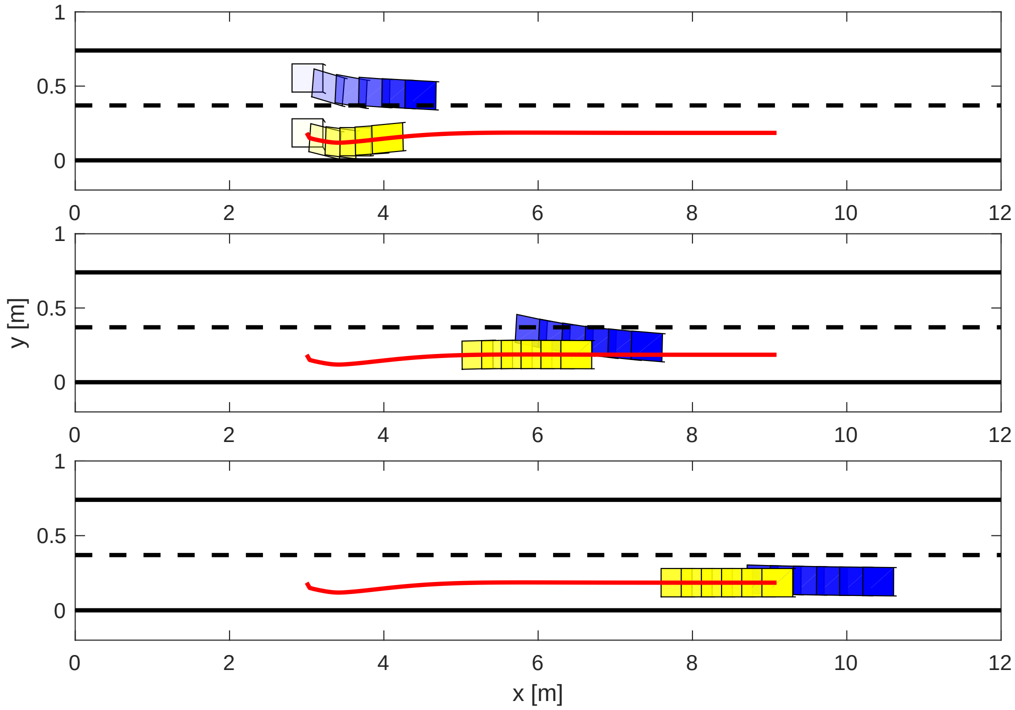

Figure 7.4: Bayesian persuasion with "tough" surrounding vehicle in lane keeping scenario

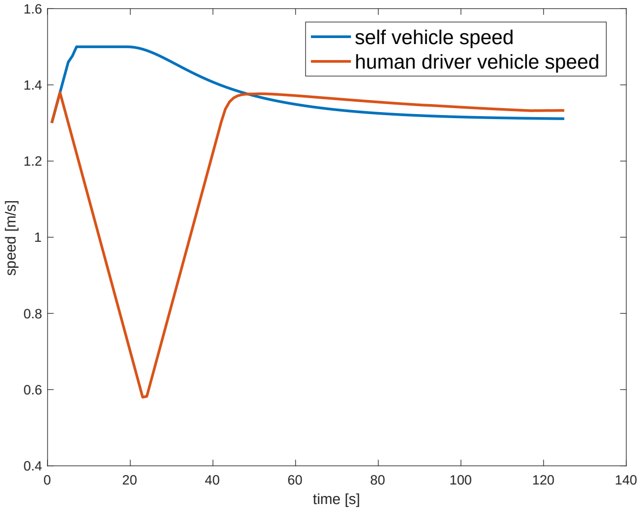

Figure 7.5: Lane keeping speed profile (persuasion with “nice” surrounding vehicle case)

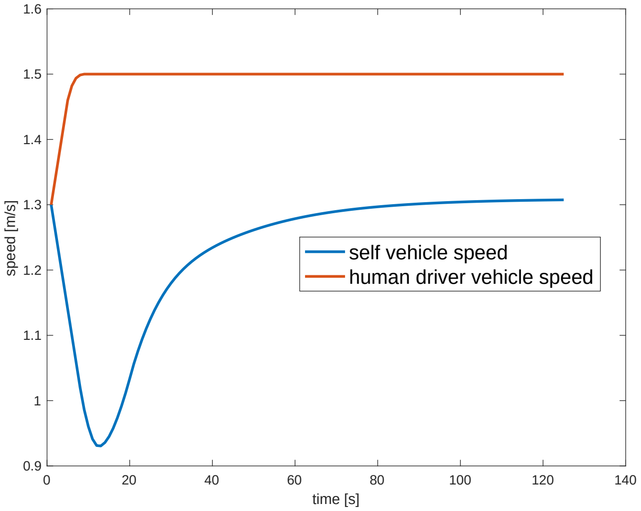

Figure 7.6: Lane keeping speed profile (persuasion with “tough” surrounding vehicle case)

Intersection Crossing Scenario

Another common driving scenario is intersection crossing, where both the vehicles need to reason about who is supposed to pass first. Fig. 7.7 and Fig. 7.8 demonstrate the algorithm's performace when interacting with different kinds of driver. In the figures, the yellow rectangle represent the ego vehicle and the blue rectangle is the surrounding vehicle. It is shown in Fig. 7.7 (Fig. 7.8) that the ego vehicle decides to pass first (second) when meeting a nice (tough) driver, which is consistent with human driving behavior. The vehicles' speed profiles are shown in Fig. 7.9 and Fig. 7.10, which further illustrate the ego vehicle's intention. In the case of nice surrounding vehicle, the ego vehicle keeps accelerating and cross the intersection first. When interacting with a tough surrounding vehicle, the ego vehicle first inches a little and then stops, waiting for the other vehicle to pass.

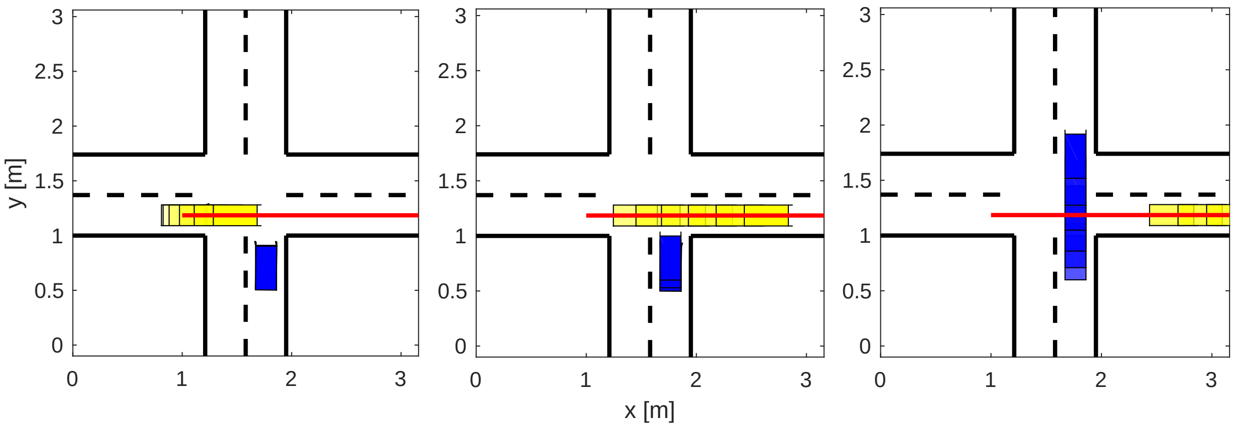

Figure 7.7: Bayesian persuasion with "nice" surrounding vehicle in intersection scenario

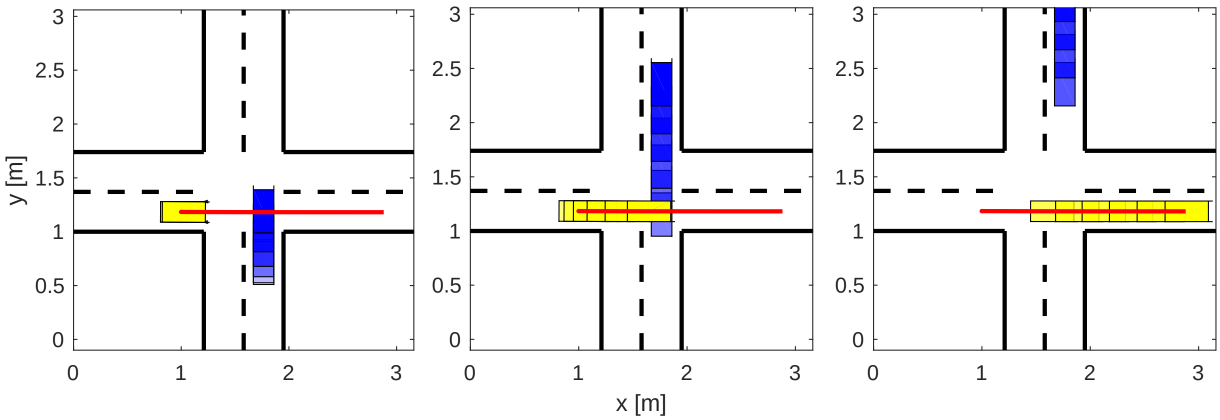

Figure 7.8: Bayesian persuasion with "tough" surrounding vehicle in intersection scenario

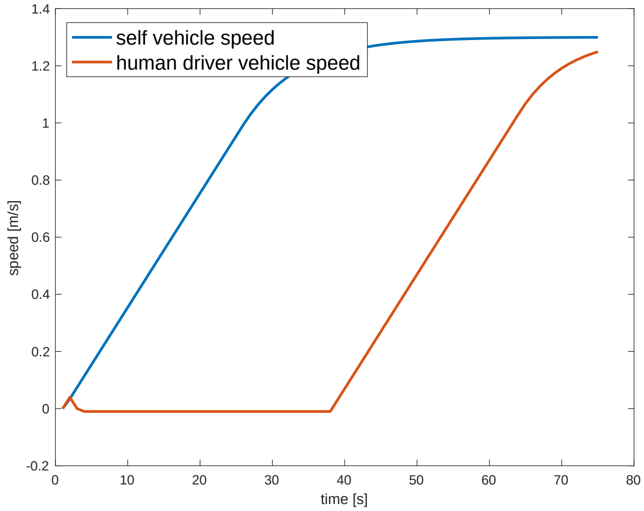

Figure 7.9: Intersection crossing speed profile (persuasion with “nice” surrounding vehicle case)

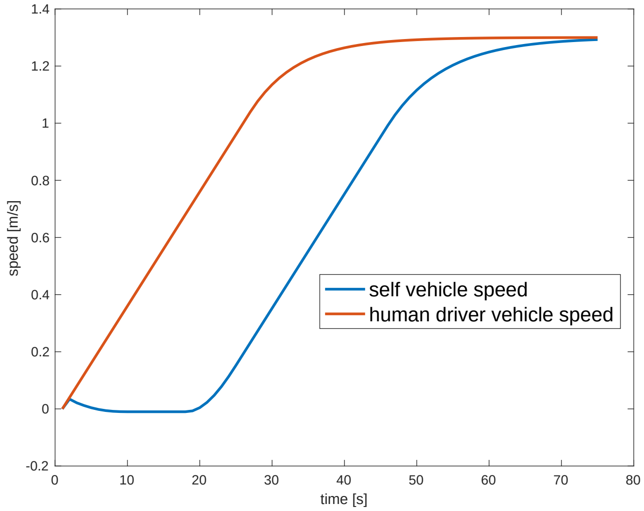

Figure 7.10: Intersection crossing speed profile (persuasion with “tough” surrounding vehicle case)

7.5 Chapter Summary

In this chapter, an optimization-based Bayesian persuasive driving algorithm was proposed. Compared to the negotiation strategy introduced in Chapter 6, by modeling the interacting vehicle's decision making process utilizing Bayesian persuasion, the intelligent interactive driving behavior could be obtained without introducing extra decision variables (the interacting vehicle's driving states for cooperative proposal). Therefore, a better computational performance could be achieved.

In the persuasion game, the ego vehicle serves as the information sender who attempts to manipulate the surrounding vehicle's (information receiver) posterior belief of the world state in order to achieve a lower cost for both players via providing information about its driving intention. The world state of the Bayesian game was defined to be the surrounding vehicle's impression about the ego vehicle. In the surrounding vehicle's point of view, both the signaling and the belief of the world state are formulated as Gaussian distributions. An integral approximation was applied to reformulate the optimization into a tractable form. As shown by simulation results in several driving scenarios, the ego vehicle is capable of interacting with various types of surrounding vehicles intelligently due to the persuasion signaling strategy.

Chapter 8

Interactive Driving Based on Persuasion Game and Intention Expression

In Chapter 6 and Chapter 7, we have developed two algorithms for the interactive driving problem including negotiation and Bayesian persuasion. Essentially, both of these approaches aim to approximate the Nash equilibrium solution of the interactive driving game via cost function and constraint set adjustment. In this chapter, we propose to formulate the problem as an explicit persuasion game between the ego vehicle and the surrounding vehicle with certain assumptions and simplification. The game's utility function is formulated using the cumulative prospect theory to achieve a more accurate and reasonable description of the surrounding vehicle's decision making process and the numerical solution to the Nash equilibrium is obtained via gradient descent approach. The Nash equilibrium solution which represents the ego vehicle's driving intention is then utilized in the ego vehicle's trajectory design module which results in the expressive and optimal driving behavior.

8.1 Introduction

A fully functioning autonomous vehicle is generally composed of three modules including perception, planning and control. If we look the autonomous vehicle as a human, the the perception module works as human's eyes and ears; control module can be treated as hands and feet and the planning module which is in charge of all the decision making tasks, commanding the vehicle to get the target without colliding with obstacles is more like the human's brain. Although these three components are equally important for the autonomous vehicle, compared to the other two, the planning module as a brain still apperas to be more complex and the problems involved are also quite challenging. Among all the planning problems, the interaction between vehicles stands out as an extremely important one which has not been fully resolved.

The ability to intelligently and safely interact with other vehicles can not only improve self driving quality but also be beneficial to the global driving environment. In the literatures, there are quite a few attempts to tackle the interactive driving problem from different points of view using different techniques. When all the involved vehicles can be controlled by one planning module and all the driving information can be accessible, a centralized control approach can be applied to find the global solution as reviewed in [8]. However, since the vehicle-to-vehicle (V2V) communication system has not been fully developed, the full information required by the centralized approach may not be accessible. Therefore, for the near future applications, the decentralized approach looks more realistic and promising. In the decentralized approach, since the other vehicles' information cannot be obtained from the communication module directly, the interacting vehicles' behavior prediction becomes necessary. Several interactive prediction methods has been proposed. For example, a partially observable Markov decision process (POMDP) based approach is proposed in [21] to solve the interactive driving problem in varying intersection driving scenarios. In [28], the other vehicle's behavior is modeled via an intelligent driver model (IDM) depending on the ego vehicle's behavior. [49] utilizes an inverse reinforcement learning algorithm to learn human drivers' cost function and then include it in the autonomous vehicle's planning optimization.

In constrast to the approaches aforementioned, in this chapter, the interactive driving problem is divided into two steps including intention decision making and intention expressive trajectory planning. With regard to the intention decisoin making, the game theory approach has shown its unique advantage in interaction handling through applications in many areas. For example, game theory is combined with psychology to solve social interaction decision mkaing problem in [9]; in [23], a game theoretic approach is introduced to achieve efficient and intelligent automated negotiation in various contexts; in [32] and [52], the game theory is applied to solve vehicle-pedestrian interaction at crosswalks and vehicle-vehicle interaction at lane changing scenario respectively. In this chapter, a two-player game is formulated for vehicle interactions in different driving scenarios, where both players have two options: yielding and passing. The game's mixed Nash equilibrium is then solved to achieve the optimal intention represented by the optimal passing probability. In order to express the intention via the ego vehicle's driving trajectory, an intention expression constraint imposed on the vehicle's speed is formulated and merged into the trajectory planninng optimization.

The remainder of the chapter is structured as follows. In Section 8.2, the interactive driving game is designed with payoff matrix dependent on vehicles' estimated driving style information and utility function formulated by the cumulative prospect theory (CPT). In Section 8.3, the formulated game's Nash equilibrium is solved to obtain the optimal intention and the corresponnding intention expression constraint is formulated in Section 8.4. In Section 8.5, the proposed algorithm's effectiveness is illustrated by simulations in lane changing and intersection crossing driving scenarios. Section 8.6 concludes the chapter.

8.2 Interactive Driving Game

In this section, the two-player game between the ego vehicle and the surrounding vehicle for the interactive driving scenario is proposed. In the game setting, each player has two possible actions: yield and pass which represent their driving intention in the current driving situation. The Nash equilibrium solution to the game is defined as the vehicles' yielding/passing probability.

Game of Chicken

The proposed persuasion game is inspired by the game of chicken which is a model of conflict for two players in game theory [47]. The game of chicken originates from a scenario where two drivers drive towards each other on a one-lane road which means that one must swerve, otherwise a collision is bound to happen and two drivers will die in the crash. If both drivers swerve, then both of them are safe. However, if only one driver swerves and the other does not, the one who swerves will be called a "chicken", meaning a coward and the iron-nerved driver will enjoy a great success.

From the game theory point of view, the game of chicken can be formulated as the payoff matrix in Table 8.1:

Table 8.1: Game of chicken

		Player Y	
		Swerve (S)	Drive (D)
Player X	Swerve (S)	$(1,1)$	$(-1,2)$
	Drive (D)	$(2,-1)$	$(-M,-M)$

According to the game description in Table 8.1, for both swerving case, both players will receive a small payoff of 1, for only-one swerving case, the swerving driver (chicken in the game) will get a penalty of -1 and the other driver will yield a higher payoff of 2. M in Table 8.1 is a large number representing a huge penalty of death for both players if they both choose to drive and a collision happens. Intuitively, in order to obtain a higher payoff in this game, a rational player is supposed to follow the principle corresponding to the game's pure Nash equilibria: if the other player chooses to swerve, then keep driving, but if the other player chooses to drive, then swerving will be the optimal action to take.

However, the practical driving scenarios can be much more complicated than the chicken game setting. First, besides the one-lane conflict situation, the most common conflict cases encountered include lane changing and intersection crossing. In order to handle the multiple situations mentioned above, the palyers' action options are generalized to yielding and passing intention corresponding to the swering and driving options in the game of chicken. Specifically, for the lane change case, the meaning of yielding is to wait and execute lane change motion later while passing means immediate start of lane changing process; for the intersection scenario, options of yielding and passing just follow their original meanings. The desired decision strategy is also replaced by the mixed Nash equilibria because of the intention's probabilistic nature.

Another challenge of the practical interactive driving game is that the driver usually does not have direct access to the other driver's intention. Therefore, the decision making is largely dependent on the analysis of the other driver's driving behavior/trajectory and the game to be implemented in the practical driving system should also be altered accordingly.

Driving Intention Estimation

In an interactive driving situation with two vehicles involved, the full scenario information is usually unaccessible which means that no one has accurate informatoin about the other vehicle's driving intention. Thus, it is necessary for the vehicle opeartors (decision making module for an autonomous vehicle) to make judgement about the interacting entity's driving intention based on the historical observation of their driving behavior before taking any action.

A general driving intention estimation mechanism based on Bayes' theorem can be formulated as:

$$\begin{aligned} p(Pass|Behavior) &= \frac{p(Behavior|Pass)p(Pass)}{p(Behavior|Yield)p(Yield)+p(Behavior|Pass)p(Pass)}, \\ p(Yield|Behavior) &= 1-p(Pass|Behavior), \end{aligned} \tag{8.1}$$

where $p(Yield)$ and $p(Pass)$ represent the prior probability of yielding and passing respectively which are the previous time step estimation results, $p(\cdot|Behavior)$ denotes the conditional probability of the intention given the driving behavior observation and similarly $p(Bahavior|\cdot)$ is the conditional probability of the observed driving behavior given the intention.

In order to achieve reliable estimation performance, the selected driving behavior should be easy to obtain from the measurement data and also strongly relevant to the vehicle's driving intention. In this chapter, for the lane changing scenario, the vehicle's predicted average speed $\hat{v}^{avg}$ is chosen as the vehicle's representative intention-reflecting drivng behavior which can be obtained based on the current speed and the most recent average acceleration or from another prediction module. Intuitively, a predicted slower vehicle is more likely to yield; a predicted faster vehicle will be reasonably more inclined to the passing option and the intention of a vehicle with predicted speed almost equal to the ego vehicle is ambiguous which means that its yielding probability and passing probability are both around 0.5. With regard to the intersection crossing case, instead of the predicted speed, we utilize the predicted time of arrival at the confict zone as the driving behavior. The predicted time of arrival $\hat{T}$ is defined as:

$$\hat{T} = \begin{cases} \frac{d^c}{\hat{v}^{avg}} & \text{if } \hat{v}^{avg} \geq \varepsilon_v \\ T_{max} & \text{if } \hat{v}^{avg} < \varepsilon_v \end{cases}, \tag{8.2}$$

where d^c is the vehicle's distance to the intersection conflict zone, T_{max} is the maximum crossing time and ε_v represents a low speed close to zero. T_{max} and ε_v are used to handle the cornercase where the vehicle is almost not moving.

In the case of lane changing, since there will be a forward/backward safety distance d_s from the interacting yielding/passing vehicle to the ego vehicle when the driving task is finished, the model of driving behavior conditioned on intention $p(Behavior|Yield)$ and $p(Behavior|Pass)$ is assumed to follow the Gaussian distributions as:

$$\begin{aligned}
&Behavior|Yield = \hat{v}_s^{avg}|Yield \sim \mathcal{N}(\hat{v}^y, \sigma_1^2),\\
&Behavior|Pass = \hat{v}_s^{avg}|Pass \sim \mathcal{N}(\hat{v}^p, \sigma_1^2),\\
&\hat{v}^y = \hat{v}_e^{avg} - \frac{d^s - d}{t_f},\\
&\hat{v}^p = \hat{v}_e^{avg} + \frac{d^s + d}{t_f},
\end{aligned} \tag{8.3}$$

where $\hat{v}_s^{avg}$ and $\hat{v}_e^{avg}$ denote the predicted average speed of the surrounding vehicle and the ego vehicle respectively, $\hat{v}^y$ and $\hat{v}^p$ represent the yielding and passing speed predicted by the vehicle, d is the current forward distance from surrounding vehicle to the ego vehicle (it will be negative if the ego vehicle is lonngitudinally behind the surrounding vehicle), d^s is the safety distance between two vehicles, t_f is the typical time of finishing the driving task, which is set to 5 seconds in this dissertatoin and σ_1 denotes the standard deviation of the Gaussian distribution.

Similarly, for the intersection case, the driving behavior model is defined as:

$$\begin{aligned}
&Behavior|Yield = \hat{T}_s|Yield \sim \mathcal{N}(\hat{T}^y, \sigma_2^2),\\
&Behavior|Pass = \hat{T}_s|Pass \sim \mathcal{N}(\hat{T}^p, \sigma_2^2),\\
&\hat{T}^y = \hat{T}_e + T^c,\\
&\hat{T}^p = \hat{T}_e - T^c,
\end{aligned} \tag{8.4}$$

where $\hat{T}_s$ and $\hat{T}_e$ are the predicted arrival time for the surrouding vehicle and the ego vehicle respectively, $\hat{T}^y$ and $\hat{T}^p$ are the predicted arrival time in the case of yielding and passing correspondingly, T^c is the typical crossing time at the intersection and σ_2 denotes the standard deviation of the distribution.

Based on the Gaussian assumptions in (8.3) and (8.4), the estimated intention probability can be calculated according to (8.1) after the predicted behavior information is revealed.

Payoff Function Design

During the driving process, as described by the game of chicken, the vehicle will always receive some reward or penalty after taking certain actions. In the real life, the reward can be a sense of satisfication for the driver, saving a little time via driving faster or avoidance of the heavy traffic at a certain lane and the possible penalty can be anger caused by other drivers, getting stuck in the traffic or an accident in the worst case.

Obviously, each driver has a different sense of reward and penalty dependent on their characteristics and lots of other unpredictable factors. To simplify the design, we assume that each vehicle's payoff is determined by the current intention of itself and the interacting vehicle estimated via (8.1). For the payoff functions design, we follow the basic idea of the chicken game and introduce the driving style information to make the outcome more realistic and human-like. The modified payoff matrix of the interactive driving game is shown in Table 8.2, where EV and SV denote the ego and surrounding vehicle, $\tilde{p}_{s|e}$ and $\tilde{p}_{e|s}$ are the passing probabilities of the surrounding vehicle estimated by the ego vehicle and the ego vehicle estimated by the surrounding vehicle respectively, p_e represents the ego vehicle's passing probability solution to the previous time step game and M is the accident penalty.

Table 8.2: Interactive driving game

		EV	
		Yield	Pass
SV	Yield	$(-2\tilde{p}_{s\vert e}/\tilde{p}_{e\vert s}, -2p_e/\tilde{p}_{s\vert e})$	$(-\tilde{p}_{s\vert e}/\tilde{p}_{e\vert s}, p_e/\tilde{p}_{s\vert e})$
	Pass	$(\tilde{p}_{s\vert e}/\tilde{p}_{e\vert s}, -p_e/\tilde{p}_{s\vert e})$	$(-M, -M)$

According to the payoff matrix, for the pass-yield pair, the passing vehicle will receive a reward and the yielding vehicle will get a penalty. Both the reward and penalty are dependent on the vehicles' passing probability. Intuitively, for a vehicle with higher passing probability, the waiting time caused by yielding will seem more unbearable which means a larger penalty. Moreover, the penalty is also influenced by the other vehicle's driving style. For example, if the other vehicle is very aggressive, then yielding will be a reasonable option resulting in a small penalty. Therefore, in the one passing and one yielding case, the penalty for the yielding vehicle is designed as proportional to the ego vehicle's current passing probability and inversely proportional to the other vehicle's. Similarly, the design of reward is based on the same reasoning. With regard to the both yielding case, since in this scenario, both vehicles will have to wait for a longer time reducing the global driving efficiency, both vehicles will get a larger penalty compared to the one vehicle yielding case. In order to reflect this effect, the yielding penalty is multiplied by a constant of 2 for both yielding case.

Remark 8.1: The passing probabilities in the payoff matrix in Table 8.2 are the prior intention probabilities before the interaction process which represent the vehicles' driving style information. In order to find the posterior optimal action probability after the interaction, the game's Nash equilibrium needs to be sovled.

Remark 8.2: In the payoff matrix, two different variables are used to represent the ego vehicle's passing probability estimated by the surrouding vehicle $\tilde{p}_{e|s}$ and the ego vehicle's previous solution p_e. The reason that these two variables are both necessary is that p_e is solved based on the prediction of the surrouding vehicle made at previous time step which is subject to change and thus can be different from the surrouding vehicle's actual new observed action. Therefore, in the surrounding vehicle's point of view, the estimated ego vehicle's intention $\tilde{p}_{e|s}$ can get misunderstood deviating from p_e. Thus, both the ego vehicle's prior real intention represented by p_e and its intention perceived by the surrounding vehicle represented by $\tilde{p}_{e|s}$ are needed for the current time step planning. The estimation of $\tilde{p}_{e|s}$ is basically the same as for $\tilde{p}_{s|e}$ with the role of ego and surrounding vehicle switched.

8.3 Solution of Nash Equilibria

In this section, the Nash equilibria of the interactive driving game defined in Table 8.2 are solved to obtain the optimal driving intentions for both vehicles. Since in most scenarios, the vehicle's intention will be neither pure yielding nor pure passing, we are more interested in the mixed equilibria instead of the pure ones.

Utility Function

In order to achieve the Nash equilibria, an utility function needs to be derived from the game payoff matrix. The most commonly used utility function is based on the statistical expectation and thus called expected utility. The expected utility theory has been successfully applied in economics, game theory and decision making area for decades. However, the theory is heavily dependent on the assumption that the decision makers are always rational and thus cannot explain some common phenomena in the decision making process especially when human players are involved. For instance, several common facts happening in human decision making that violate the standard expected utility model are listed in the following:

- Nonlinear preferences: It has been observed that people's preference in different options is not linear in the corresponding outcome probability. For example, as demonstrated by the Allais paradox experiment connducted by Maurice Allais in 1953 [2], the difference between probabilities of 0.99 and 1.0 has more influence on people's selected choice then the difference between 0.1 and 0.11 although both differences have the same value of 0.01.

- Risk seeking: Although in many applications of decision making, it is assumed that people tend to avoid risk when uncertainties are present, however, there are actually lots of observations of risk-seeking behavior in decision making problems. For example, people usually prefer winning a large reward with a small probability over following the ”correct” option given by the expected value.

- Loss aversion: In the situation with both uncertainty and risk, the decision makers often choose to avoid loss instead of pursuing a better gain. This asymmetry is also hard to explain using the expected utility theory.

In order to explain these common phenomena in human players' decision making process and achieve a more realistic utility function that is in line with people's psychological thinking, the cumulative prospect theory (CPT) is applied. The cumulative prospect theory was first propsoed by Tversky and Kahneman in 1992 [55] and provides another model for descriptive decisions under risk and uncertainty. The two basic ideas of CPT are 1) mapping the original payoff function of the game to a value function that has different effect on gains and losses to address loss aversion effect and 2) transformation of the objective cumulative probability to a subjective cumulative weighting curve to model people's nonlinear preferences of probabilities. The mapped wieghting function is supposed to be nonsensitive to moderate probabilities, overestimate small probabilities and underestimate high probabilities. These two mapping of values and weights will jointly influence the decision makers' risk seeking behavior.

The general requirement for the value function mapping is that it should be concave for gains, convex for losses and steeper for losses than for gains. An examplar value function mapping is as

$$v(x)=\begin{cases} x^{\alpha} & \text{if } x\geq 0 \\ -\lambda(-x)^{\beta} & \text{if } x<0 \end{cases}, \tag{8.5}$$

where x represents the original payoff and $\alpha>0$, $\beta>1$, $\lambda\geq 1$ are the model parameters predefined. In this dissertation, we choose the same values as estimated in the original CPT paper [55], which are $\alpha=\beta=0.88$ and $\lambda=2.25$.

In order to reflect people's tendecy of overestimating extreme but unlikely events and underestimating average events, the mapping from the cumulative objective probability to subjective perceived weights can be expressed as:

$$\begin{aligned} w^{+}(cp) &= \frac{cp^{\gamma}}{(cp^{\gamma}+(1-cp)^{\gamma})^{1/\gamma}}, \\ w^{-}(cp) &= \frac{cp^{\delta}}{(cp^{\eta}+(1-cp)^{\eta})^{1/\eta}}, \end{aligned} \tag{8.6}$$

where cp is the objective cumulative probability, w^{+} and w^{-} are mapping functions for events with loss and gain respectively. The mapping parameters γ and η are positive scalars between 0 and 1, which are selected as $\gamma=0.61$ and $\eta=0.69$ in this dissertation, again following the estimation in the original CPT paper [55]. The relationship between the objective cumulative probability and the mapped weights are shown in Fig. 8.1, which clearly shows the effect of small probability overweighting and high probability underweighting.

With the value function and probability mapping defined in (8.5) and (8.6) respectively, the utility function U based on CPT is formulated as:

$$\begin{aligned} & U=U^{+}+U^{-}, \quad U^{+}=\sum_{i=0}^{n}\pi_i^{+}v(x_i), \quad U^{-}=\sum_{i=-m}^{0}\pi_i^{-}v(x_i), \\ & \pi_i^{+}=w^{+}(p_i^E+\cdots+p_n^E)-w^{+}(p_{i+1}^E+\cdots+p_n^E), \\ & \pi_i^{-}=w^{-}(p_{-m}^E+\cdots+p_i^E)-w^{-}(p_{-m}^E+\cdots+p_{i-1}^E), \\ & \pi_n^{+}=w^{+}(p_n^E), \quad \pi_{-m}^{-}=w^{-}(p_{-m}^E), \end{aligned} \tag{8.7}$$

where x is the original payoff, i is the event index and p_i^E is the probability of event i happening. The negative i's represent index for loss event, the positive ones are for gain events and the indexing can be in any order. Particularly, for the interactive driving game, the event probability matrix is shown in Table 8.3, where p_e^a and p_s^a denote the action passing probability for the ego and surrounding vehicle respectively.

By combining the payoff matrix in Table 8.2, the event probability matrix in Table 8.3 and the utility formulation in (8.7), the utility of the ego and surrounding vehicle can be expressed as $U_e(p_e^a,p_s^a)$ and $U_s(p_e^a,p_s^a)$ respectively, which are functions of p_e^a and p_s^a.

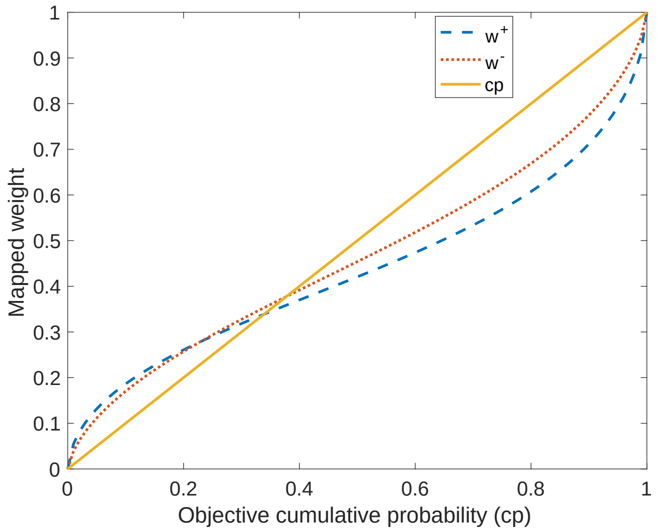

Figure 8.1: Mapping from objective cumulative probability to subjective weight

Table 8.3: Interactive driving game event probabilities

		EV	
		Yield	Pass
SV	Yield	$(1-p_e^a)(1-p_s^a)$	$p_e^a(1-p_s^a)$
	Pass	$(1-p_e^a)p_s^a$	$p_e^a p_s^a$

Numerical Solution Strategy

After achieving the utility functions $U_e(p_e^a, p_s^a)$ and $U_s(p_e^a, p_s^a)$, the interactive driving game is fully defined and we are ready to find the game's mixed Nash equilibrium, i.e., the optimal pair of action probabilities $(p_e^{a,*}, p_s^{a,*})$. The resulting optimization problem can be formulated as a two-step procedure:

- Step 1: Solve for the optimal $p_s^{a,*}$ as a function of p_e^a:

$$p_s^{a,*}(p_e^a) = \arg\max_{p_s^a} U_s(p_e^a, p_s^a). \tag{8.8}$$

- Step 2: Solve for the optimal $p_e^{a,*}$:

$$p_e^{a,*} = \arg\max_{p_e^a} U_e(p_e^a, p_s^{a,*}(p_e^a)). \tag{8.9}$$

However, since the utility functions $U_e(p_e^a, p_s^a)$ and $U_s(p_e^a, p_s^a)$ are highly nonlinear due to the CPT formulation in (8.7), the analytical form of $p_s^{a,*}(p_e^a)$ is hard to obtain. Therefore, a numerical solution strategy is developed to solve for $(p_e^{a,*}, p_s^{a,*})$. The procedure is described as:

- Step 1: Sample for p_e^a between 0.05 and 0.95 meaning that the extreme events are eliminated. Each sampling point is denoted as $p_{e,k}^a$.
- Step 2: For each p_e^a sampling point at $p_{e,k}^a$, solve for the corresponding optimal $p_{s,k}^{a,*}$ as

  $$p_{s,k}^{a,*} = \arg\max_{p_s^a} U_s(p_{e,k}^a, p_s^a).$$

  As $U_s(p_{e,k}^a, p_s^a)$ is still a highly nonlinear and complicated function with regard to p_s^a, more efforts are required to solve the problem. Note that p_s^a only appears in the mapped weighting terms and both mapping functions w^+ and w^- are differentiable. Therefore, the gradient descent approach is applied to get the numerical solution $p_{s,k}^{a,*}$.
- Step 3: Calculate the value of U_e for each pair of candidate solution $(p_{e,k}^a, p_{s,k}^{a,*})$. The candidate pair with maximum U_e value is selected as the optimal solution $(p_e^{a,*}, p_s^{a,*})$.

8.4 Trajectory Planning with Intention Expression

In this section, the obtained optimal action probability pair is merged into the trajectory design module so that the ego vehicle can effectively express its updated driving intention and intelligently react to the surrounding vehicle's new action. The trajectory planning optimization problem is first defined and the intention expression is formulated as a convex constraint.

The standard model predictive control (MPC) based optimization problem for vehicle trajectory planning is formulated as:

$$\begin{aligned}\min_{\mathbf{u}_{t|t_0}} \quad & \sum_{t=t_0}^{t_0+H} \|\mathbf{x}_{t|t_0} - \mathbf{x}_{des}\|_Q^2 + \sum_{t=t_0}^{t_0+H-1} \|\mathbf{u}_{t+1|t_0} - \mathbf{u}_{t|t_0}\|_R^2 \\ & (x_{t|t_0}, y_{t|t_0}) \in \mathbb{S}(x_s^p(t), y_s^p(t)), \\ & \mathbf{u}_{t|t_0} \in \mathbb{U}, \\ & \mathbf{x}_{t+1|t_0} = f(\mathbf{x}_{t|t_0}, \mathbf{u}_{t|t_0}), \\ & \mathbf{x}_{t_0|t_0} = \mathbf{x}(t_0),\end{aligned} \tag{8.10}$$

where t_0 is the current time instant, H is the receding horizon length, $\mathbf{x} = [x, y, \theta, v]^T$ is the vehicle's state vector including its x, y positions, yaw angle θ and speed v, $\mathbf{u} = [a, \delta]^T$ is the control input for the vehicle including acceleration a and steering angle δ and $f(\mathbf{x}, \mathbf{u})$ represents the vehicle dynamics model. $\mathbf{x}_{des}$ denotes the vehicle's desired state, $\mathbf{x}_{t|t_0}$ and $\mathbf{u}_{t|t_0}$ denote the state and control input vector at t predicted at the current time instant t_0 respectively and $\mathbf{x}(t_0)$ is the vehicle's initial state. $\mathbb{S}(x_s^p(t), y_s^p(t))$ denotes the safety set defined by the surrounding vehicle's predicted t time instant (x, y) positions $(x_s^p(t), y_s^p(t))$ and $\mathbb{U}$ is the feasible set for the vehicle's control input. Q and R are positive definite penalty matrices for deviation from desired state and control change respectively.

In order to express the optimal action intention via the driving trajectory, the intention expression needs to be combined into the optimization framework. In fact, the intention expression is a reverse process of the intention estimation. For the estimation purpose, the other vehicle's behavior observation is utilized as the input and the estimated intention probability is the desired output. In the expression process, however, the input is the optimal intention and the aim is to find the appropriate driving behavior which the other vehicle can understand and therefore a smooth interaction process can be achieved. The intention expression can be formulated as the following constraint:

$$p_e^{a,l} \leq p(Pass|Behavior) \leq p_e^{a,*}, \tag{8.11}$$

where

$$p_e^{a,l} = max(p_e^{a,*} - \varepsilon_p, 0.01)$$

is the lower bound for the acceptable passing intention probability interval and ε_p is a small probability. The value of 0.01 is used to prevent the lower bound from going to zero. Based on the intention estimation formula (8.1), the intention expression constraint (8.11) can be rewritten as:

$$\frac{p_e^{a,l}}{1-p_e^{a,l}} \leq \frac{p(Behavior|Pass)p(Pass)}{p(Behavior|Yield)p(Yield)} \leq \frac{p_e^{a,*}}{1-p_e^{a,*}}, \tag{8.12}$$

where $p(Pass) = \tilde{p}_{e|s}$ and $p(Yield) = 1 - p(Pass)$ are the ego vehicle's current passing and yielding probability before interaction estimated by the surrounding vehicle, which are both known variables.

Similarly with the estimation process, the predicted average speed of the vehicle and the predicted time of arrival are selected as the representative driving behavior for the lane changing and intersection crossing scenario respectively. The surrounding vehicle's predicted average speed $\hat{v}_s^{avg}$ can be extracted from the prediction module and the ego vehicle's can be formulated as:

$$\hat{v}_e^{avg} = \frac{1}{H}\sum_{t=t_0}^{t_0+H} v_{t|t_0}. \tag{8.13}$$

Then the predicted time of arrival $\hat{T}_e$ can be developed accordingly based on the definition in (8.2).

Since the behavior model is assumed to be Gaussian in (8.3), the probability $p(Behavior|Pass)$ and $p(Behavior|Yield)$ can be expanded as:

$$p(Behavior|Pass) = \frac{1}{\sqrt{2\pi\sigma_1^2}}exp(-\frac{(\hat{v}_e^{avg} - \hat{v}_s^{p})^2}{2\sigma_1^2}),$$

$$p(Behavior|Yield) = \frac{1}{\sqrt{2\pi\sigma_1^2}}exp(-\frac{(\hat{v}_e^{avg} - \hat{v}_s^{y})^2}{2\sigma_1^2}),$$

where $\hat{v}_s^p$ and $\hat{v}_s^y$ are the passing and yielding speed predicted by the surrounding vehicle following the definition in (8.3). Then the explicit form of the intention expression constraint (8.12) can be derived for the lane chaning scenario as:

$$\frac{2\sigma_1^2}{\hat{v}_s^p - \hat{v}_s^y} ln\left(\frac{p_e^{a,l}(1-\hat{p}_{e|s})}{\hat{p}_{e|s}(1-p_e^{a,l})}\right) \leq \frac{2}{H}\sum_{t=t_0}^{t_0+H} v_{t|t_0} - \hat{v}_s^p - \hat{v}_s^y \leq \frac{2\sigma_1^2}{\hat{v}_s^p - \hat{v}_s^y} ln\left(\frac{p_e^{a,*}(1-\hat{p}_{e|s})}{\hat{p}_{e|s}(1-p_e^{a,*})}\right), \tag{8.14}$$

which is a linear constraint imposed on the ego vehicle's speeds in the receding horizon. Similarly, the intention expression constraint for the intersection scenario can be developed as:

$$\frac{2d_e^c}{\hat{T}_s^p + \hat{T}_s^y + \frac{2\sigma_2^2 ln\left(\frac{p_e^{a,*}(1-\hat{p}_{e|s})}{\hat{p}_{e|s}(1-p_e^{a,*})}\right)}{\hat{T}_s^p - \hat{T}_s^y}} \leq \frac{1}{H}\sum_{t=t_0}^{t_0+H} v_{t|t_0} \leq \frac{2d_e^c}{\hat{T}_s^p + \hat{T}_s^y + \frac{2\sigma_2^2 ln\left(\frac{p_e^{a,l}(1-\hat{p}_{e|s})}{\hat{p}_{e|s}(1-p_e^{a,l})}\right)}{\hat{T}_s^p - \hat{T}_s^y}}, \tag{8.15}$$

where d_e^c is the ego vehicle's distance to the intersection conflict zone, $\hat{T}_s^p$ and $\hat{T}_s^p$ are the yielding and passing arrival time predicted by the surrounding vehicle.

By adding the intention expression constraint (8.14) for the lane changing scenario or (8.15) for the intersection crossing scenario to the original trajectory planning optimization (8.10), the expressive trajectory can be achieved for the ego vehicle so that the surrounding vehicle is able to get a better a understanding of the driving situation. Therefore, the interaction process will be more efficient and more intelligent.

8.5 Simulation Results

In this section, the proposed interactive driving game approach's performance is demonstrated in two scenarios which are lane changing and intersection crossing. An MPC optimization approach is applied to the surrounding vehicles. Different driving styles are simulated by different safety weights, where the small weight represents the aggressive driver and the large safety weight means the relatively conservative or friendly driver. The setting details for the simulation are the same as in Table 6.1 and Table 6.2. In the simulations, the yellow rectangles represent the ego vehicle and green, blue rectangles are the surrounding vehicles.

Lane Changing

In the lane changing scenario, the ego vehicle needs to interact with two vehicles in sequence. The first vehicle is always aggressive meaning that it will not yield to the ego vehicle. Both the aggressive and friendly driving styles are simulated for the second surrounding vehicle. Fig. 8.2 shows the interaction result with the friendly driver, where the ego vehicle got to merge into the target lane before the second surrounding vehicle. The aggressive second surrounding vehicle case simulation result is shown in Fig. 8.3, where the ego vehicle could not find a chance to merge until the second surrounding vehicle passed. The corresponding speed profiles for the friendly and aggressive interacting vehicle cases are shown in Fig. 8.4 and Fig. 8.5 respectively, which illustrate the interacting vehicle's yielding and passing process.

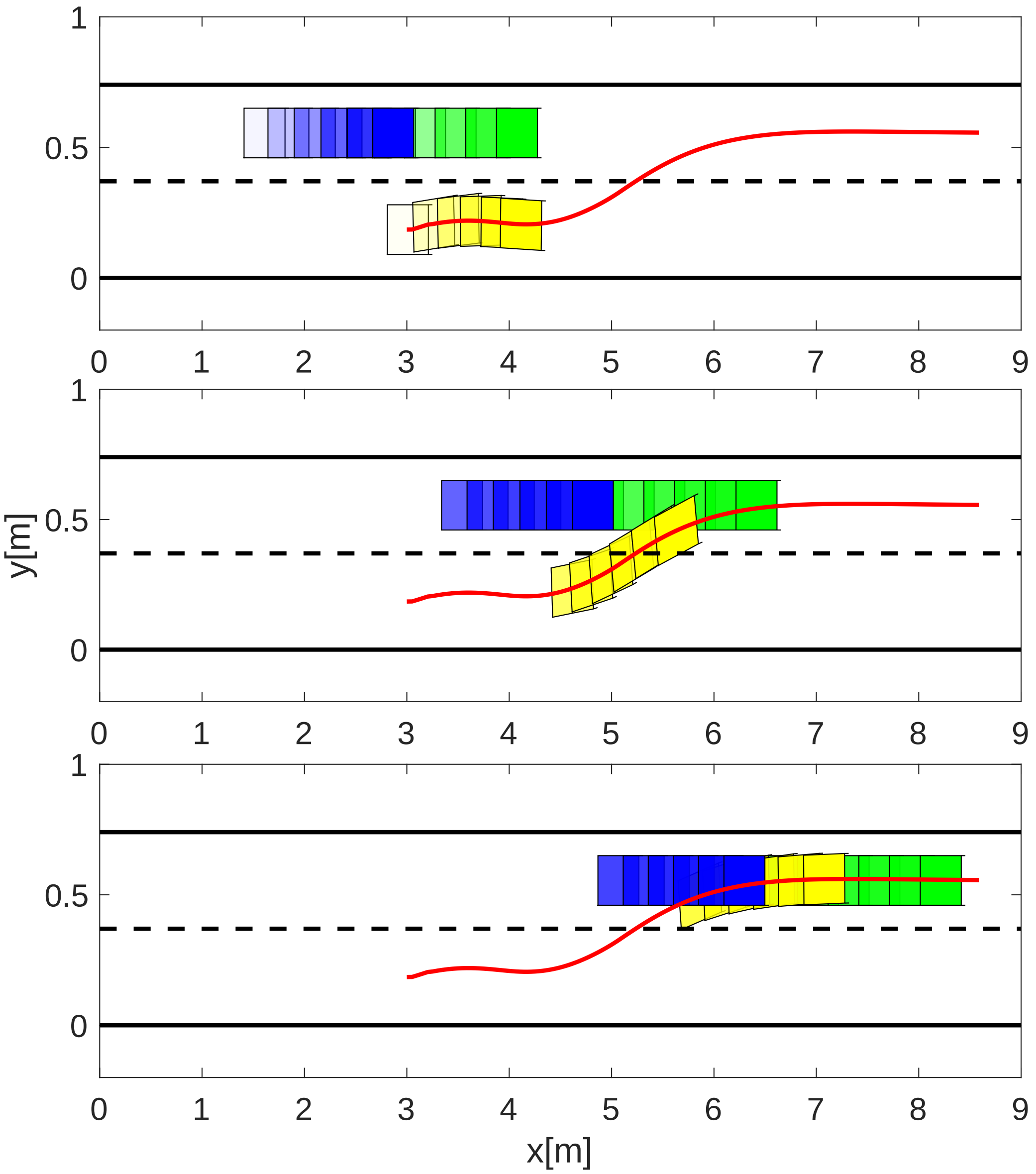

Figure 8.2: Game playing with "friendly" surrounding vehicle2 in lane change scenario

Intersection Crossing

With regard to the intersection crossing scenario, two vehicles are assumed to arrive the intersection around the same time and there is no priority difference between them. In this scenario, since both vehicles prefer passing first to save waiting time and they also need to avoid collision, the

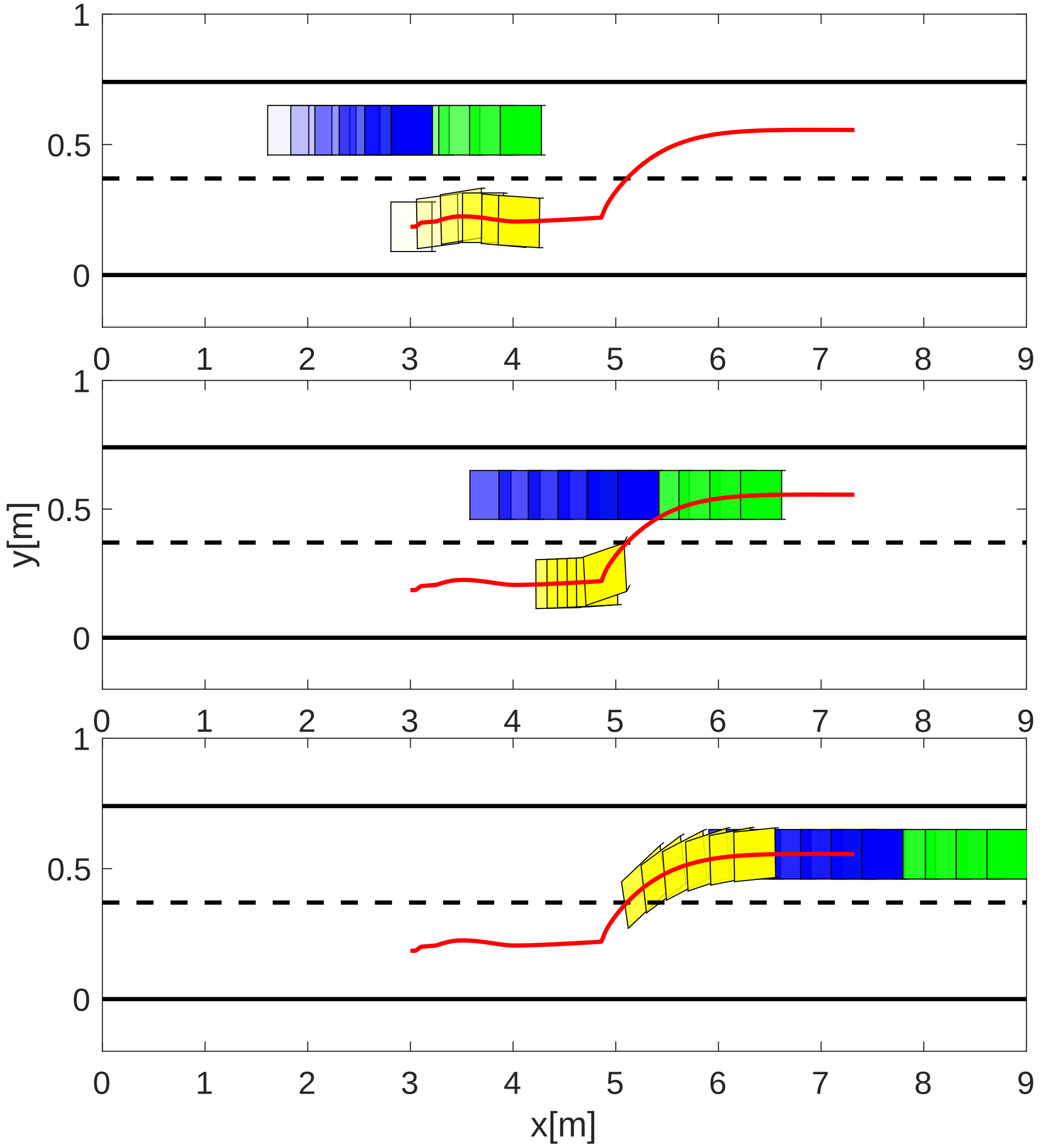

Figure 8.3: Game playing with “aggressive” surrounding vehicle2 in lane change scenario

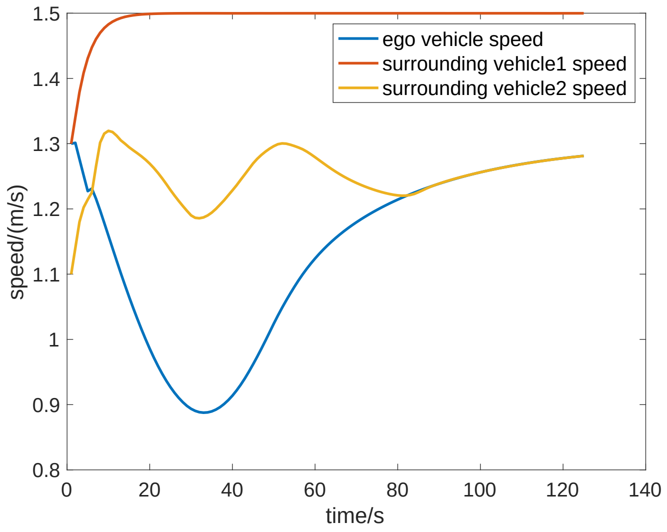

Figure 8.4: Lane change speed profile (game with “friendly” surrounding vehicle case)

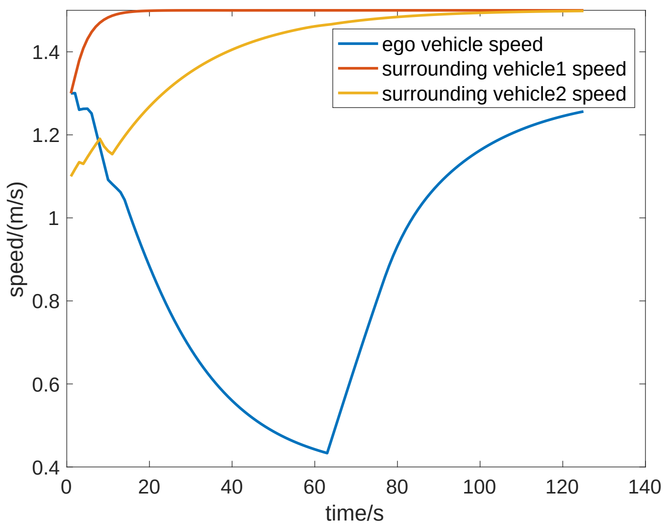

Figure 8.5: Lane change speed profile (game with “aggressive” surrounding vehicle case)

interaction influence needs to be taken into account for both vehicles' decision making process, which can be fulfilled by the interactive driving game approach proposed. The simulated ego vehicle and friendly surrounding vehicle interacting scenario is shown in Fig. 8.6, where the interacting surrounding vehicle decided to yield and the ego vehicle crossed the intersection first. The speed profile for both vehicles in this scenario is shown in Fig. 8.7. Another exemplar case where the ego vehicle needs to interact with an aggressive vehicle is shown in Fig. 8.8, where the surrounding vehicle chose the passing first option which forced the ego vehicle to wait. The corresponding speed profile is given in Fig. 8.9.

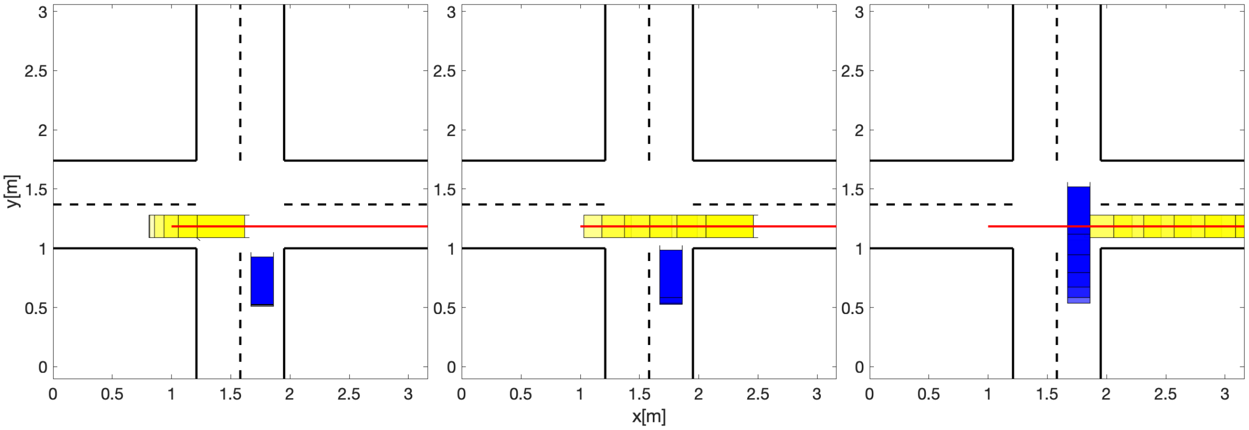

Figure 8.6: Game playing with "friendly" surrounding vehicle in intersection scenario

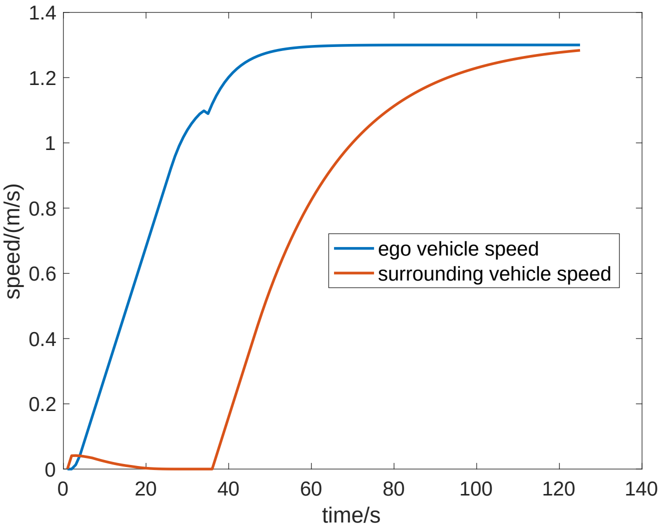

Figure 8.7: Intersection crossing speed profile (game with "friendly" surrounding vehicle case)

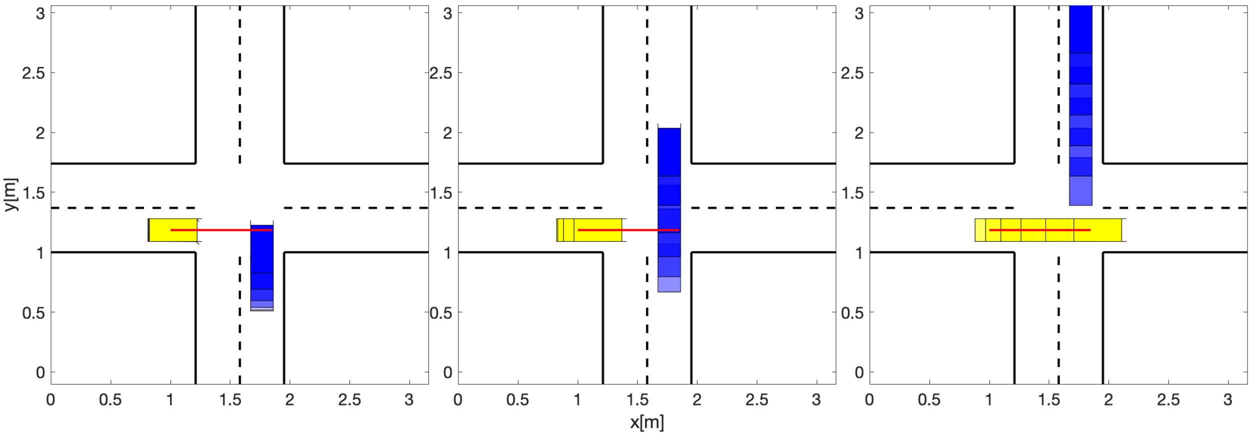

Figure 8.8: Game playing with “aggressive” surrounding vehicle in intersection scenario

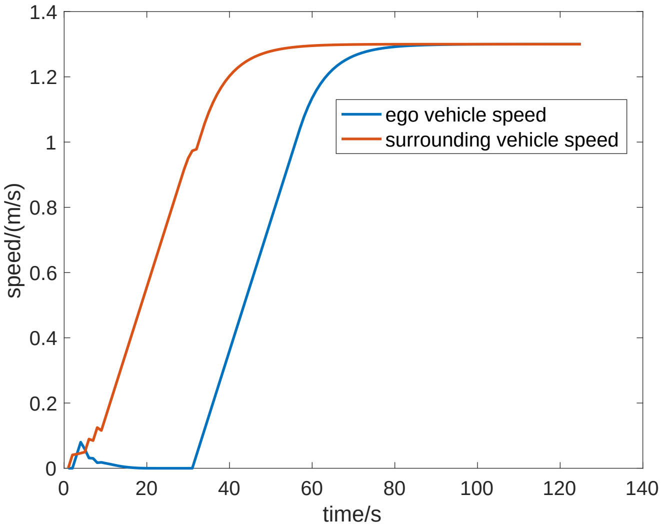

Figure 8.9: Intersection crossing speed profile (game with “aggressive” surrounding vehicle case)

8.6 Chapter Summary

In this chapter, the interactive driving problem was formulated as a two-player game between the ego vehicle and the surrounding vehicle. The game’s payoff matrix was designed based on both vehicles’ current driving intention before interaction process represented by their passing probabilities estimated by each other. The intention estimation was achieved using Bayesian method. In order to combine the loss aversion and nonlinear preference phenomena which are common in people’s natural decision making process, the cumulative prospect theory (CPT) was applied to define the game’s utility function. The game’s Nash equilibrium, which is the optimal action passing probability pair for both vehicles, was solved by a numerical solution strategy based on gradient descent. The obtained optimal intention was then formulated as a linear intention expression constraint imposed on the ego vehicle's speed in the trajectory planning optimization module. The proposed approach's performance was verified by simulations in the lane changing and intersection crossing scenarios.

Compared to the interactive driving algorithms proposed in Chapter 6 and Chapter 7, the persuasion game approach explicitly solved for the Nash equilibrium solution representing the optimum in interactive scenarios. However, the algorithm performance would be more dependent on the payoff and cost function structure.

Chapter 9

Conclusion and Future Work

9.1 Summary

In this dissertation, three control structures for the multi-agent system with interaction were explored including centralized, distributed and decentralized control. The introduced algorithms were applied to solve interactive agents control problems under two application backgrounds: intelligent building and autonomous driving.

Intelligent Building System

In Part I, the intelligent building thermal system was studied. The thermal dynamic model identification was achieved in Chapter 2 which built the foundation for controller design. A lumped first-order resistance and capacitance (RC) model was utilized to describe the building thermal dynamics. The model estimation was based on the semiparametric regression approach which separates the model into two parts: a parametric part representing the RC model structure and a non-parametric part corresponding to the un-modeled complex thermal effects. For each part, an autoregressive-moving average (ARMA) time series model was trained. The identified model was then rewritten into the finite time horizon form using lifting technique.

Chapter 3 proposed a centralized iterative controller design approach to compensate the heat interaction effect between zones and reject disturbances. The developed control system was composed of a feedforward iterative learning controller (ILC) and an iteratively tuning feedback controller. The ILC was responsible for eliminating the influence of repetitive disturbances caused by repeating human activities and weather pattern. The feedback controller was introduced to enhance the system's robustness against non-repetitive disturbance components. To fulfill the energy saving objective and satisfy zone occupants' thermal comfort demand, the controller design was formulated as an optimization with to-be-designed controllers as decision variables. The optimization solution provided an optimal feedforward and possible non-causal feedback controller combination. A stabilizing projection step was then proposed to find the causal feedback controller. Due to the projection step in the iterative controller design algorithm, the optimality of feedforward and feedback controller combination did not hold anymore. In Chapter 4, a centralized concurrent controller design algorithm was introduced to achieve the optimal combination of feedforward ILC and causal implementable feedback controller. In the proposed algorithm, Youla-parametrization technique was applied so that the feedback controller designed was guaranteed to be stabilizing. By introducing new variables, the controller design optimization was convexified and both controllers' solutions could be achieved simultaneously by one optimization step.

As the centralized approach will fail for large-scale systems, in Chapter 5, a distributed cooperative ILC algorithm was proposed for the multi-agent building system with communication between agents enabled. In the distributed control system, all the agents share the same global cost function. In order to achieve systemwise cooperation, the distributed system was designed with two basic functions: solving for local controllers and broadcasting the updated controller results. For each agent, with the information received from the interacting entities, its corresponding local controller was solved by a distributed optimization. Afterwards, the solution results would be broadcast to the communication system so that the other agents could adjust their control decisions accordingly. After repeating such procedure for iterations, the convergence to global optimum could be reached. The algorithm's performance was illustrated by simulations and the convergence property was theoretically proven.

Autonomous Driving System

In Part II of the dissertaion, several decentralized algorithms were proposed for the autonomous driving system. In Chapter 6, the interaction between vehicle agents was formulated as an optimization based negotiation about the right of the road. The two key factors of the negotiation, namely, concession and persuasion were both introduced into the optimization framework to achieve intelligent driving behaviors. The concession was realized by a tuning weight in the cost function making tradeoff betwen the ego vehicle and interacting vehicles. The persuasion was expressed by a convex constraint imposed on the ego vehicle's driving state so that it could actively claim its own driving right. The concession and persuasion were tuned according to the interacting vehicle's reaction to the ego vehicle's cooperative proposal.

In the negotiation algorithm, the ego agent needed to plan for the interacting party as well to make a proposal, which increased the computatinal load. In Chapter 7, a Bayesian persuasion based algorithm was developed to reduce the computational load. In the Bayesian persuasion, the ego vehicle agent was chosen as the persuader providing information and the interacting vehicle was the corresponding information receiver. Since the interacting agent's decision making was dependent on the world state influenced by the persuader, by providing appropriate information, the persuader would be able to manipulate the interacting agent's posterior belief of the driving situation and thus exert influence on its future behavior. The persuasion process was formulated as an optimization. With the Gaussian assumption and integral approximation, the problem was transformed to a convex form.

In Chapter 8, the agents' interaction was formulated as a two-player persuasion game, which was developed from the game of chicken. In the persuasion game, both players had two options: yielding and passing and the game's mixed Nash equilibrium represented interacting agents' optimal yielding/passing probabilities in the current driving scenario. The game's payoff matrix was designed based on the agent's driving style information estimated by each other. In order to imitate human's decision making strategy, the cumulative prospect theory (CPT) was applied to develop the utility function from the payoff matrix. Since the utility in CPT form was highly nonlinear, a numerical solution was found for the Nash equilibrium. The obtained optimal intention was then formulated as a convex constraint in the trajectory planning optimization. With the optimal intention decision made by the persuasion game and expressive trajectory optimization, the ego autonomous vehicle was capable of performing human-like behavior in various interaction scenarios.

9.2 Future Work

This dissertation only serves as one step towards ultimate intelligent autonomous agent, there are still many open questions for the interaction challenge including cost function design, joint interaction among multiple agents and computational efficiency.

Cost Function Design

In this dissertation, the controllers were desgined using an optimization based approach minimizing the cost function subject to certain constraints. However, since there are no general principles for the cost function design, the definition of cost terms are still empirical dependent on the designer's knowledge and understanding of the system, which is vulnerable to biases and cognitive limitations. Moreover, some factors in the decision making process are hard or even impossible to be quantitatively described, e.g., in the driving system, the excitement of overtaking some other vehicles and pressure when surrounded by some huge trucks. Therefore, establishing a general systematic and objective methodology for quantitative cost function design is critical for next generation interactive autonomous agent control.

Joint Interaction Between Multiple Agents

In a multi-agent system, it is quite possible that three or more agents will interact with each other at the same time. Such joint interaction process can be properly handled using the centralized and distributed algorithm proposed for the building system in this dissertation. However, it can be very troublesome for the decentralized approach. In order to consider influence of more other agents, the ego agent needs to increase its own optimization problem size and takes into account the estimation inaccuracy exponentially increased. In addition, the two-agent interaction solution algorithm may require fundamental structure change to be applicable for joint interaction problems.

Computational Efficiency

The computational efficiency is critical for any algorithm to be applied in the real engineering system. In this dissertation, all the formulated optimization problems are convexified so that they are valid for online implementatoin. However, along with the increase of problem size and number of interacting agents, more efficient optimization solving technique becomes necessary. Furthermore, for some engineering systems, the convexification may not be realizable or the control performance will be significantly degraded after relaxation. For such systems, efficient non-convex optimization solving algorithms will be more favorable.

Bibliography

[1] A. Afram and F. Janabi-Sharifi. “Theory and applications of HVAC control systems-A review of model predictive control (MPC)”. In: *Building and Environment* 72 (2014), pp. 343–355.

[2] M. Allais. “Le comportement de l’homme rationnel devant le risque: critique des postulats et axiomes de l’école américaine”. In: *Econometrica: Journal of the Econometric Society* (1953), pp. 503–546.

[3] A. Aswani et al. “Identifying models of HVAC systems using semiparametric regression”. In: *2012 American Control Conference (ACC)*. IEEE. 2012, pp. 3675–3680.

[4] D. A. Bristow, M. Tharayil, and A. G. Alleyne. “A survey of iterative learning control”. In: *IEEE Control Systems* 26.3 (2006), pp. 96–114.

[5] B. Bueno et al. “A resistance-capacitance network model for the analysis of the interactions between the energy performance of buildings and the urban climate”. In: *Building and Environment* 54 (2012), pp. 116–125.

[6] C. G. Cena et al. “A cooperative multi-agent robotics system: Design and modelling”. In: *Expert Systems with Applications* 40.12 (2013), pp. 4737–4748.

[7] C. Chen and S. Peng. “Learning control of process systems with hard input constraints”. In: *Journal of Process Control* 9.2 (1999), pp. 151–160.

[8] L. Chen and C. Englund. “Cooperative intersection management: A survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.2 (2015), pp. 570–586.

[9] A. M. Colman. “Cooperation, psychological game theory, and limitations of rationality in social interaction”. In: *Behavioral and brain sciences* 26.2 (2003), pp. 139–153.

[10] G. R. De Campos, P. Falcone, and J. Sjoberg. “Autonomous cooperative driving: a velocity-based negotiation approach for intersection crossing”. In: *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE. 2013, pp. 1456–1461.

[11] M. C. De Oliveira, J. Bernussou, and J. C. Geromel. “A new discrete-time robust stability condition”. In: *Systems & control letters* 37.4 (1999), pp. 261–265.

[12] A. D. Dragan and S. S. Srinivasa. *Formalizing assistive teleoperation*. MIT Press, July, 2012.

[13] U. Endriss. “Monotonic concession protocols for multilateral negotiation”. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM. 2006, pp. 392–399.

[14] *EnergyPlus website*. URL: `http://apps1.eere.energy.gov/buildings/energyplus/cfm/weather`.

[15] C. E. Garcia, D. M. Prett, and M. Morari. “Model predictive control: theory and practice–a survey”. In: *Automatica* 25.3 (1989), pp. 335–348.

[16] S. Gunnarsson and M. Norrlof. “On the design of ILC algorithms using optimization”. In: *Automatica* 37.12 (2001), pp. 2011–2016.

[17] J. K. Gupta, M. Egorov, and M. Kochenderfer. “Cooperative multi-agent control using deep reinforcement learning”. In: *International Conference on Autonomous Agents and Multiagent Systems*. Springer. 2017, pp. 66–83.

[18] M. R. Hafner et al. “Automated vehicle-to-vehicle collision avoidance at intersections”. In: *Proceedings of world congress on intelligent transport systems*. 2011.

[19] M. R. Hafner et al. “Cooperative collision avoidance at intersections: Algorithms and experiments”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013), pp. 1162–1175.

[20] A. Hock and A. P. Schoellig. “Distributed iterative learning control for a team of quadrotors”. In: *Conference on Decision and Control (CDC)*. 2016, pp. 4640–4646.

[21] C. Hubmann et al. “Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction”. In: *IEEE Transactions on Intelligent Vehicles* 3.1 (2018), pp. 5–17.

[22] C. Hubmann et al. “Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles”. In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE. 2017, pp. 1671–1678.

[23] N. R. Jennings et al. “Automated negotiation: prospects, methods and challenges”. In: *Group Decision and Negotiation* 10.2 (2001), pp. 199–215.

[24] E. Kamenica and M. Gentzkow. “Bayesian persuasion”. In: *American Economic Review* 101.6 (2011), pp. 2590–2615.

[25] A. Kelman and F. Borrelli. “Bilinear model predictive control of a HVAC system using sequential quadratic programming”. In: *IFAC World Congress*. Vol. 18. 2011, pp. 9869–9874.

[26] B. Lautenschlager and G. Lichtenberg. “Data-driven Iterative Learning for Model Predictive Control of Heating Systems”. In: *IFAC-PapersOnLine* 49.13 (2016), pp. 175–180.

[27] J. H. Lee, K. S. Lee, and W. C. Kim. “Model-based iterative learning control with a quadratic criterion for time-varying linear systems”. In: *Automatica* 36.5 (2000), pp. 641–657.

[28] D. Lenz, T. Kessler, and A. Knoll. “Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 447–453.

[29] D. Lenz et al. “Deep neural networks for Markovian interactive scene prediction in highway scenarios”. In: *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE. 2017, pp. 685–692.

[30] Y. Ma and F. Borrelli. “Fast stochastic predictive control for building temperature regulation”. In: *American Control Conference (ACC)*. 2012, pp. 3075–3080.

[31] D. Meng and K. L. Moore. “Robust cooperative learning control for directed networks with nonlinear dynamics”. In: *Automatica* 75.1 (2017), pp. 172–181.

[32] A. Millard-Ball. “Pedestrians, autonomous vehicles, and cities”. In: *Journal of Planning Education and Research* 38.1 (2018), pp. 6–12.

[33] M. Minakais, S. Mishra, and J. T. Wen. “Groundhog Day: Iterative learning for building temperature control”. In: *IEEE International Conference on Automation Science and Engineering (CASE)*. 2014, pp. 948–953.

[34] M. Minakais et al. “Iterative Learning Control for Coupled Temperature and Humidity in Buildings”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 13420–13425.

[35] L. Monostori, J. Váncza, and S. R. T. Kumara. “Agent-based systems for manufacturing”. In: *CIRP annals* 55.2 (2006), pp. 697–720.

[36] P. D. Morosan et al. “Building temperature regulation using a distributed model predictive control”. In: *Energy and Buildings* 42 (2010), pp. 1445–1452.

[37] F. Oldewurtel et al. “Energy efficient building climate control using stochastic model predictive control and weather predictions”. In: *Proceedings of the 2010 American control conference*. IEEE. 2010, pp. 5100–5105.

[38] C. Peng, L. Sun, and M. Tomizuka. “Distributed and cooperative optimization-based iterative learning control for large-scale building temperature regulation”. In: *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2017, pp. 1606–1611.

[39] C. Peng and M. Tomizuka. “Bayesian Persuasive Driving”. In: *American Control Conference (ACC)*. IEEE. 2019, accepted.

[40] C. Peng and M. Tomizuka. “Concurrent Design of Feedforward and Feedback Controller for Building Thermal System”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 5257–5263.

[41] C. Peng and M. Tomizuka. “Cooperative Driving Based on Negotiation with Persuasion and Concession”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1118–1124.

[42] C. Peng, W. Zhang, and M. Tomizuka. “Iterative design of feedback and feedforward controller with input saturation constraint for building temperature control”. In: *American Control Conference (ACC)*. 2016, pp. 1241–1246.

[43] C. Peng et al. “Optimization-based constrained iterative learning control with application to building temperature control system”. In: *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2016, pp. 709–715.

[44] Z. Peng et al. “Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory”. In: *International Journal of Systems Science* 46.8 (2015), pp. 1447–1457.

[45] L. Prez-Lombard, J. Ortiz, and C. Pout. “A review on buildings energy consumption information”. In: *Energy and Buildings* 40.3 (2008), pp. 394–398.

[46] R. Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[47] A. Rapoport and A. M. Chammah. “The game of chicken”. In: *American Behavioral Scientist* 10.3 (1966), pp. 10–28.

[48] A. Rosenfeld and S. Kraus. “Strategical Argumentative Agent for Human Persuasion.” In: *ECAI*. 2016, pp. 320–328.

[49] D. Sadigh et al. “Planning for Autonomous Cars that Leverage Effects on Human Actions.” In: *Robotics: Science and Systems*. 2016.

[50] M. Sandipan, U. Topcu, and M. Tomizuka. “Optimization-based constrained iterative learning control”. In: *IEEE Transactions on Control Systems Technology* 19.6 (2011), pp. 1613–1621.

[51] L. Sun, X. Chen, and M. Tomizuka. “Selective iterative learning control with non-repetitive disturbance rejection”. In: *Proceedings of 2014 International Symposium on Flexible Automation*. July 2014.

[52] A. Talebpour, H. S. Mahmassani, and S. H. Hamdar. “Modeling lane-changing behavior in a connected environment: A game theory approach”. In: *Transportation Research Procedia* 7 (2015), pp. 420–440.

[53] B. Thiesson et al. “ARMA time-series modeling with graphical models”. In: *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press. 2004, pp. 552–560.

[54] K. Toohey and M. Duckham. “Trajectory similarity measures”. In: *SIGSPATIAL Special* 7.1 (2015), pp. 43–50.

[55] A. Tversky and D. Kahneman. “Advances in prospect theory: Cumulative representation of uncertainty”. In: *Journal of Risk and uncertainty* 5.4 (1992), pp. 297–323.

[56] S. N. Venkat et al. “Distributed MPC strategies with application to power system automatic generation control”. In: *IEEE Transactions on Control Systems Technology* 16.6 (2008), pp. 1192–1206.

[57] D. Youla, H. Jabr, and J. Bongiorno. “Modern Wiener-Hopf design of optimal controllers–Part II: The multivariable case”. In: *IEEE Transactions on Automatic Control* 21.3 (1976), pp. 319–338.

[58] R. Zheng et al. “Automated multilateral negotiation on multiple issues with private information”. In: *INFORMS Journal on Computing* 28.4 (2016), pp. 612–628.