

DATA-DRIVEN MODELING AND CONTROL OF NONLINEAR AND COMPLEX
SYSTEMS WITH APPLICATION ON AUTOMOTIVE SYSTEMS

By

Kaian Chen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2022

ABSTRACT

DATA-DRIVEN MODELING AND CONTROL OF NONLINEAR AND COMPLEX SYSTEMS WITH APPLICATION ON AUTOMOTIVE SYSTEMS

By

Kaian Chen

As data is becoming more readily accessible from various systems, data-driven modeling and control approaches have gained increased popularity among both researchers and practitioners. Compared to its first-principle counterparts, data-driven approaches require minimal domain knowledge and calibration efforts, which is especially appealing to nonlinear and complex systems. In this thesis, two efficient data-driven frameworks, *indirect* and *direct*, for the control of nonlinear and complex systems are presented.

The *indirect* method first involves an efficient online system identification approach with a composite local model structure. We introduce the concept of evolving Spatial-Temporal Filters (eSTF) that dynamically transforms an incoming input-output data stream into a nonlinear combination of local models. Each local model is assigned with an ellipsoid-shaped cluster that is used to define its validity zone, and a distance metric that combines the Mahalanobis distance to the clusters and the scaled local model prediction error is exploited to compute the local model composition weights. The cluster and model parameters are efficiently updated online using input-output data stream, enabling adaptive system identification with efficient computations.

With the identified eSTF model structure, we then develop an efficient quasi linear parameter varying (qLPV) based stochastic model predictive controller (SMPC) for a class of nonlinear systems subject to chance constraints and additive disturbance. The qLPV form is established with the scheduling variable and a set of linear time-invariant models obtained from the eSTF system identification approach. To handle chance constraints, probabilistic reachable sets – the probabilistic analogy of robust reachable sets – are exploited to tighten

the constraints to robustly guarantee constraint satisfaction despite model uncertainties and additive disturbances.

A shifted scheduling variable strategy is designed such that the resultant MPC optimization can be efficiently solved by solving a series of quadratic programming problems. The *indirect* data-driven modeling and control pipeline is successfully applied to automotive powertrain systems with great system identification and control performance demonstrated.

On the other hand, we further develop a *direct* data-driven control paradigm that leverages behavioural system theory, and directly generates control commands from input/output data without the need of a parametric model. We exploit singular value decomposition (SVD)-based order reduction to significantly reduce the online computation complexity without degrading the control performance. This control paradigm is successfully applied to battery fast charging, which has complex dynamics and is difficult to model. Furthermore, the *direct* data-driven approach heavily relies on the intensive collection and sharing of data, which raises serious privacy concerns, especially for systems with multiple agents. Therefore, we develop a privacy-preserving data-enabled predictive control scheme where we exploit affine-masking to protect the privacy of shared input/output data. It is then applied to the control of connected and automated vehicles (CAVs) in a mixed traffic environment with promising results demonstrated through comprehensive simulations.

Copyright by
KAIAN CHEN
2022

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ALGORITHMS	x
KEY TO ABBREVIATIONS	xi
CHAPTER 1 INTRODUCTION	1
1.1 Motivation	1
1.2 Indirect Method vs. Direct Method	1
1.3 Research Goals	2
1.4 Contributions	3
1.5 Organization of Thesis	4
CHAPTER 2 NONLINEAR SYSTEM IDENTIFICATION WITH AN EVOLVING SPATIAL-TEMPORAL FILTERS	5
2.1 Background - Internal Combustion Engine	5
2.2 Introduction	7
2.3 Problem Statement	9
2.4 Spatial-Temporal Filtering for Local Model Interpolation	11
2.5 Online Identification of Composite Linear Local Models	17
2.5.1 System Reformulation	17
2.5.2 Recursive Least Squares	18
2.5.3 Towards Implementation	19
2.6 Application to Turbo-Charged Engine System Identification	20
2.6.1 Simulation-based Validation	20
2.6.2 Experimental Vehicle Data Validation	22
2.7 Conclusion	25
CHAPTER 3 STOCHASTIC MODEL PREDICTIVE CONTROL FOR QUASI LINEAR PARAMETER VARYING SYSTEMS	26
3.1 Introduction	26
3.2 Problem Formulation	29
3.3 Stochastic MPC for Quasi LPV Model	31
3.3.1 Chance Constraint Handling	32
3.3.2 qLPV-SMPC with Indirect Feedback	33
3.4 Problem Reformulation for Efficient qLPV-SMPC	36
3.5 Case Study on Automotive Powertrain Control	40
3.5.1 Pseudo-Plant Construction From Vehicle Data	41
3.5.2 Performance Index and System Constraints	41
3.5.3 Simulation Results of qLPV-SMPC on NA Engine Model	43
3.5.4 Comparison between qLPV-RMPC and qLPV-SMPC	45

3.6	Conclusion	47
CHAPTER 4 EFFICIENT AND PRIVACY-PRESERVING DATA-ENABLED PREDICTIVE CONTROL FOR AUTOMOTIVE SYSTEMS		
4.1	Introduction of Data-Enabled Predictive Control	49
4.2	Preliminaries	50
4.3	Efficient DeePC for Lithium-Ion Battery Fast Charging	54
4.3.1	Introduction of Lithium-ion Battery Fast Charging Problem	54
4.3.2	DeePC Formulation for Fast Charging of Lithium-ion Battery	58
4.3.2.1	Working Conditions for Lithium-ion Battery Charging	58
4.3.2.2	Non-Parametric Representation of Lithium-Ion Battery Charging Dynamics	59
4.3.2.3	DeePC Formulation for Battery Fast Charging	60
4.3.3	DeePC with Dimension Reduction	61
4.3.4	Simulations	63
4.3.4.1	Experimental Setup	63
4.3.4.2	Performance of DeePC for Battery Fast Charging	64
4.3.4.3	Performance of DeePC with Dimension Reduction	66
4.3.5	Conclusion	69
4.4	Privacy-preservation based DeePC for Automated Vehicle Control in Mixed Traffic Environment	69
4.4.1	Introduction of DeePC for Automated Vehicle Control in Mixed Traffic Environment	69
4.4.2	System Model of Mixed Traffic	73
4.4.3	Data-Enabled Predictive Leading Cruise Control	76
4.4.3.1	Non-Parametric Representation of Mixed Traffic	76
4.4.3.2	<i>DeeP-LCC</i> Formulation	78
4.4.4	Privacy-Preserving <i>DeeP-LCC</i>	79
4.4.4.1	Attack Model	79
4.4.4.2	Affine Masking	81
4.4.4.3	Privacy-Preserving <i>DeeP-LCC</i> Reformulation	82
4.4.5	Equivalence and Privacy Preservation	85
4.4.5.1	Equivalence with Affine Transformation	86
4.4.5.2	Privacy Preservation	87
4.4.6	Numerical Experiments	89
4.4.6.1	Scenario A: Comprehensive Acceleration and Deceleration	90
4.4.6.2	Scenario B: Emergency Braking	92
4.4.7	Conclusion	94
BIBLIOGRAPHY		95

LIST OF TABLES

Table 2.1	Parameters for engine system identification with a high-fidelity engine simulator.	20
Table 3.1	Comparison of total fuel consumption of qLPV-RMPC and qLPV-SMPC. . . .	46
Table 4.1	Safety constraint values.	64
Table 4.2	DeePC performance index based on A vs. $\bar{A}$	68
Table 4.3	Fuel Consumption and Average Absolute Velocity Error (AAVE) in Scenario A.	92

LIST OF FIGURES

Figure 2.1	Schematic diagram of turbo-charged internal combustion engine.	5
Figure 2.2	Performance comparison between STF and Neural Network model (one-step prediction).	22
Figure 2.3	One model generation timing with aggressive ECAM advancing at open throttle and retarded ICAM.	23
Figure 2.4	One-step fuel rate prediction on validation data.	23
Figure 2.5	Coefficients of sub-models.	24
Figure 2.6	Multi-step prediction performance.	24
Figure 3.1	Scheduling variable and qLPV model updating strategy.	38
Figure 3.2	Torque tracking performance.	44
Figure 3.3	State error $e(k)$ distribution during the whole MPC optimization process and the PRS $\mathcal{R}^x$ with $p_x = 0.85$	44
Figure 3.4	A segment of MPC optimization process. Top: a segment of 3D PRS tube and corresponding system trajectories; Left-Bottom: Projection onto $x_1 - t$ plane; Right-Bottom: Projection onto $x_2 - t$ plane. The blue circle shows a constraint violation point.	45
Figure 3.5	Comparison of tube size: mRPIS vs. PRS.	46
Figure 3.6	Comparison between qLPV-RMPC and qLPV-SMPC: torque tracking.	47
Figure 3.7	Comparison between qLPV-RMPC and qLPV-SMPC: control trajectory.	47
Figure 4.1	Schematic diagram of the data-enabled predictive control.	51
Figure 4.2	Simulation results of DeePC and CCCVs.	65
Figure 4.3	DeePC prediction vs real system measurement. (a). Comparison between $y_{1 t}$ and $y(t+1), t = 0, 1, 2, \dots$. (b). Comparison between $[y_{1 \bar{t}}, \dots, y_{10 \bar{t}}], \bar{t} = 0, 10, 20, \dots$ and $y(t+1), t = 0, 1, 2, \dots$	66
Figure 4.4	Simulation results of original DeePC and SVD-based DeePC.	67
Figure 4.5	Cost comparison between SVD-based DeePC and original DeePC in the case of same Hankel matrix size.	68
Figure 4.6	Schematic of mixed traffic system with $n + 1$ vehicles.	73

Figure 4.7	Unsecure <i>DeeP-LCC</i> architecture.	80
Figure 4.8	Privacy-preservation <i>DeeP-LCC</i> architecture.	85
Figure 4.9	Velocity profiles of the mixed traffic system under Scenario A. (a) All the vehicles are HDVs. (b) <i>DeeP-LCC</i> under the Hankel matrix structure is utilized to generate the control for CAVs. (b) Privacy-preserving <i>DeeP-LCC</i> under the Hankel matrix structure is utilized to generate the control for CAVs.	91
Figure 4.10	Velocity profiles of the mixed traffic system under Scenario B. (a) All the vehicles are HDVs. (b) <i>DeeP-LCC</i> under the Hankel matrix structure is utilized to generate the control for CAVs. (b) Privacy-preserving <i>DeeP-LCC</i> under the Hankel matrix structure is utilized to generate the control for CAVs.	92
Figure 4.11	Simulation results with privacy-preserving <i>DeeP-LCC</i> under Scenario B: (a) actual state and input information of CAVs, (b) state and input information exchanged between the CAVs and the central unit.	93

LIST OF ALGORITHMS

Algorithm 2.1	Online nonlinear system identification with spatial-temporal filters. .	16
Algorithm 2.2	RLS model parameter update.	19
Algorithm 3.1	Quasi LPV Stochastic MPC.	40

KEY TO ABBREVIATIONS

eSTF	evolving Spatial-Temporal Filters
ANN	Artificial Neural Network
ICAM	intake cam position
ECAM	exhaust cam position
ARX	autoregressive model with exogenous input
NARX	nonlinear autoregressive model with exogenous input
CDF	cumulative distribution function
RLS	recursive least squares
PRML	pseudo-random multilevel
FTP	federal test procedure
NEDC	new European driving cycle
BFR	best fitting rate
DOE	design of experiment
PCM	power-train control module
LPV	linear parameter-varying
qLPV	quasi LPV
MPC	model predictive controller
SMPC	stochastic MPC
RMPC	robust MPC
NMPC	nonlinear MPC
QP	quadratic programming
LTI	linear time invariant
PRS	probabilistic reachable set
NA	natural aspirated
NRMSE	normalized root-mean-square-error

mRPIS minimal robust positively invariant set

DeePC data-enabled predictive control

PE persistent excitation

EV electronic vehicle

Li-ion lithium-ion

CC constant current

CV constant voltage

EECM electric equivalent circuit model

EM electrochemical model

SOC state-of-charge

V2I vehicle-to-infrastructure

V2V vehicle-to-vehicle

CACC cooperative adaptive cruise control

CAV connected and automated vehicle

HDV human-driven vehicle

LCC leading cruise control

OVN optimal vehicle model

ADP adaptive dynamic programming

CHAPTER 1

INTRODUCTION

1.1 Motivation

Modern industrial systems are getting increasingly complex and often evolve over time with dynamics changing. Therefore, controlling such systems is a challenging task. One example is modern automotive systems [1], whose development is driven by the advancement of electrical control units, coupled mechanical structure, and all kinds of software, making it highly complex and nonlinear. The classical approach to control such systems is to first obtain a physical model by using physics-based modeling methods [2], e.g., thermodynamics, Newton's law, and hydromechanics, followed by extensive experiments for model validation. The controller design and synthesis are then performed based on the calibrated model. This model-based process provides a good system understanding, however, much expert efforts, tedious parameters tuning and model reduction/simplification are needed for practical application. Unmodeled dynamics sometimes can also cause practical issues. As such, data-driven control strategies [3] are getting more attention and research interests, which circumvent the aforementioned cost and efforts in physics-based approaches and only require data collection that is relatively easy in most cases. The increasing system complexity in modern engineering systems also motivates the development of data-driven control approaches.

1.2 Indirect Method vs. Direct Method

Two forms of data-driven methods for controller design are presented in this thesis, the indirect method and the direct method, while data plays the central role in both methods. For the indirect method, a data-driven modeling [4], [5] step is conducted first to identify a dynamical model, based on which a controller is synthesized. Its advantage is that the minimal domain knowledge is required, because all processes are only based on measurable

system data, no intricate physics are involved for both system modeling and controller design. However, this method leads to the lack of physical understanding of the underlying model and good system identification algorithm is critical to guarantee the accuracy of the developed model. With the same advantage, but in contrast to the indirect method, the direct method does not rely on any parametric or explicit models [3], the input-output data is directly used for controller design. The intermediate system identification step is avoided in the direct method by leveraging the behavioral system theory [6] and considering the system trajectories implicitly. Both methods can be used towards an optimization-based problem for real-world applications.

1.3 Research Goals

The principal objective of this thesis is to develop data-driven control approaches that can efficiently control systems while satisfying system constraints, e.g., physical limitation, and actuator dynamics. Based on the indirect and direct methods, the objective is accomplished with the following two research aims.

The first research aim is to design an indirect method based optimal control framework. Specifically, a data-driven online/offline nonlinear system identification method is developed to obtain an accurate description of the underlying system in a simple, efficient and stable manner. Then with the developed system model, a model-based predictive control strategy is designed for a general class of nonlinear system with chance constraints and stochastic system uncertainties, i.e., modeling mis-match, estimation error and external disturbances.

In contrast to the indirect method, the second research aim is to utilize a direct method for controller design to achieve the aforementioned objective. To facilitate its online application, approaches are developed to improve its efficiency. In addition, to protect its privacy in data sharing, a privacy-preserving strategy is developed for privacy-sensitive applications.

1.4 Contributions

The main contributions of this thesis are listed as follows:

1. A novel data-driven nonlinear system identification framework capable of online adaptation is developed for identifying system dynamics [7], [8]. An evolving clustering strategy is developed for model updating and new model generation. A versatile structure of the algorithm can work with arbitrary local interpolation, e.g. focal point, linear model, Markov chain, neural network.
2. The proposed system identification method is applied to Ford testing vehicle for online engine dynamic system modeling [8], [9]. The algorithm works in an online fashion where it simultaneously updates the clusters and the local models with a training data stream.
3. Based on the identified mathematical description of the target nonlinear system, a stochastic MPC in linear parameter varying structure is developed with explicit consideration of system uncertainty, subject to chance constraints on both system state and control input [10].
4. A pilot demonstration of the proposed stochastic MPC on automotive powertrain system for multi-objective control proves its online efficiency and promising result is achieved.
5. A direct control method, data-enabled predictive control (DeePC), is formulated for lithium-ion (Li-ion) battery fast charging application and its online computational efficiency is improved by a singular value decomposition (SVD) based dimension reduction method [11].
6. A privacy-preserving DeePC framework is developed for privacy protection, demonstration is conducted via Leading Cruise Control (LCC) application [12].

1.5 Organization of Thesis

The subsequent chapters of this thesis are organized as follows:

Chapter 2: Nonlinear System Identification with Evolving Spatial-Temporal Filters [8]

This chapter introduces the proposed data-driven nonlinear system identification method - evolving Spatial-Temporal Filters (eSTF). A problem formulation of this method is introduced that is capable of online adaptation. An implementation on real vehicle engine system is demonstrated with promising results.

Chapter 3: Stochastic Model Predictive Control for Quasi Linear Parameter Varying Systems [10]

In this chapter, the considered target system is nonlinear with unknown system dynamics. A quasi Linear Parameter Varying (qLPV) structure with respect to this nonlinear system is transformed into the proposed eSTF method. Based on this structure, an efficient qLPV-MPC is developed and formulated with explicit consideration of additive stochastic system disturbance, and subject to system chance constraints on both state and control input. A pilot demonstration of qLPV-SMPC is applied on a high-fidelity engine simulation platform.

Chapter 4: Efficient and Privacy-Preserving Data-Enabled Predictive Control [11], [12]

In this chapter, an efficient data-enabled predictive control (DeePC) method is introduced. In addition, as data sharing is privacy sensitive, a privacy-preserving method is also developed on DeePC for privacy protection without impacting the controller performance. Two real-world applications are described in details, including Li-ion battery fast charging control and fleet control of connected and automated vehicles in a mixed traffic.

CHAPTER 2

NONLINEAR SYSTEM IDENTIFICATION WITH AN EVOLVING SPATIAL-TEMPORAL FILTERS

2.1 Background - Internal Combustion Engine

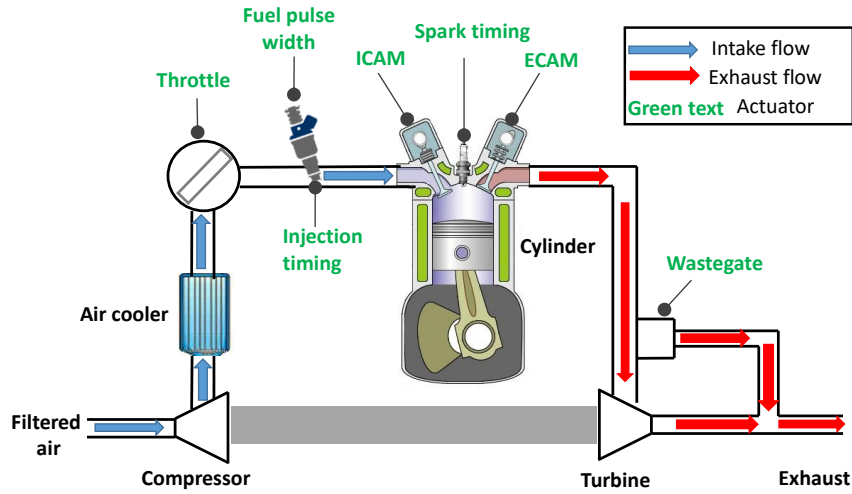


Figure 2.1 Schematic diagram of turbo-charged internal combustion engine.

The schematic diagram of the internal combustion (IC¹) engine is illustrated in Figure 2.1 where a turbine is equipped to increase the system efficiency and power output by forcing more air into the combustion chamber. This type of engine is referred to as turbo-charged engines. If the engine system is not boosted by a turbine and aspirates air into the combustion chamber naturally, it is called naturally-aspirated (NA) engine. The biggest difference between the two types of engines lies in whether a turbine exists or not. Either has its own advantages and disadvantages. Turbo-charged engine enables smaller size and higher efficiency. However, compared to NA engine, the lag between turbine spooling up and throttle response causes noncontinuous power delivering, and therefore generally leading to jerky accelerations. Here we take the turbo-charged engine as an introduction example, the working

¹This thesis only focuses on IC engine system, without any confusion, 'IC' will be omitted.

principle of NA engine is basically the same except for no turbine system existed, i.e., an exhaust re-circulation system, including wastegate valve as shown in fig. 2.1.

For turbo-charge engine system, there are seven main actuators: throttle position, fuel pulse width, intake cam position (ICAM), exhaust cam position (ECAM), fuel injection timing, spark timing, and wastegate position. The throttle and wastegate control the air charge to engine intake manifold. More specifically, the throttle controls the air flow to intake manifold; larger throttle opening leads to more air into manifold. The wastegate controls the bypass exhaust flow to regulate the turbine speed, which subsequently regulates the boost pressure. The smaller opening of wastegate valve, the more exhaust flow driving the turbine, and the higher the boost pressure, and therefore the more power the engine can generate (note that NA engine does not contain the boosting step). The intake cam (ICAM) and exhaust cam (ECAM) are used to control the gas exchange between the intake and exhaust with the cylinders and have significant (and complicated) impacts on the engine operations. Other than air charge, another important ingredient to generate engine torque is fuel. The fuel can be injected to the intake manifold, (see Figure 1) or can be injected directly into the combustion chamber. The mass ratio of the air over fuel, the so called air-to-fuel ratio, is critical to combustion process and therefore the engine efficiency and emission. The amount of fuel injection is controlled by the fuel pulse width and fuel pressure. The fuel pulse width is defined as the amount of time the fuel injector stays open during an intake cycle. Higher fuel pulse width corresponds to longer fuel injection and consequently more fuel. The injection timing is the piston position at which the injector starts to inject fuel. It also has great impact on the engine operation but the relationship to power and emission is challenging to quantify [13]. Another important factor to engine combustion is the spark timing, which refers to the piston position at which the spark ignites the mixed air and fuel.

2.2 Introduction

In this chapter, we develop a data-driven system identification method that exploits observed input-output data to develop mathematical models of the underlying dynamic system. This method is appealing as engine data can be conveniently and cost-effectively collected from engine dynamometers and/or experimental vehicles. While theories, methodologies, and algorithms for linear system identification have been well established [14], nonlinear system identification is still a demanding topic to be addressed for many real-world nonlinear and complex systems such as powertrain system [15], chemical process [16], and mechanical systems [17].

Nonlinear system identification of turbo-charged engines using black box models such as artificial neural network (ANN) have found some success, see e.g., [18]–[21]. The main drawback of ANN models is the lack of transparency, that is, they provide little physical knowledge of the underlying system. Furthermore, ANN models are typically incapable of extrapolating, requiring a comprehensive training dataset that covers the entire input-output space. In addition, their applicability to real time system identifications is rather limited.

Alternatively, composite local model approaches have been developed to deal with system nonlinearities [22]–[24]. The idea is to build local models that are good approximations in different operating regimes, and then compose a global model by appropriately interpolating these local models. It is worth noting that the well-known Takagi-Sugeno fuzzy model [25] is a special case of composite local model where the local models are linear and local models are interpolated based on fuzzy membership functions.

The main challenge in composite local model approaches lies in choosing an adequate model structure, which specifies the validity regime of each local model and an interpolation scheme to compose the global model. For instance, in the case of Takagi-Sugeno fuzzy system, the model structure refers to the design and placement of membership functions, which typically requires in-depth system insights and great tuning efforts. Several studies have been conducted to identify appropriate model structures. For example, in [26], a clustering

algorithm is exploited to identify the focal points in fuzzy membership functions. However, the use of Euclidean distance metric has limited capability of dealing with multi-dimensional skewed data sets. In [22], a logistic discriminant tree is used to partition the input space and compose the local models. However, this partition scheme provides little system insights and local model prediction accuracy is not considered in the partition. An optimization-based multicategory discrimination approach is developed in [27] to partition the input space for piecewise linear modeling. However, only input space is considered in the partitioning.

In this chapter, we develop a novel Spatial-Temporal Filter (STF) approach for engine modeling that dynamically maps input-output data to a vector of weights on local models. An evolving ellipsoidal-shape clustering algorithm is exploited to identify the filters where the proposed dissimilarity metric consists of a Mahalanobis distance and the scaled local model prediction error, similar to the idea in [28] that defines dissimilarity metric using both prediction error and parameter similarity. The Mahalanobis distance captures the spatial distance between a new data point and the existing clusters, while the scaled prediction error quantifies the prediction performance of local models. The combination of these two terms provides insights on how close the new data point is to the existing cluster/model combos and determine which cluster the new data point should be used to update or a new cluster should be created. A softmax-like function is applied on the dissimilarities to generate the interpolating weights that can be used to produce the composite model as a weighted sum of local models. Filter parameters and local model parameters are identified simultaneously online with great efficiency. The proposed algorithm is applied to the multiple-input-multiple-output turbo-charged engine modeling with promising performance demonstrated with vehicle data. Compared to our conference paper [7], the work presented here focuses on the application to the engine system, with experimental data for identification and validation. Furthermore, we have also added detailed discussions on practical implementation of the algorithm, such as the parameter interpretation and selection.

The contributions presented in this chapter include the following. Firstly, we develop

an online nonlinear system identification algorithm with composite local model structure by using spatial-temporal filters. The spatial-temporal filters integrate unsupervised learning (evolving clustering) and supervised learning (recursive least squares), and map input-output data to a vector of local model weights. Secondly, a dissimilarity metric that combines Mahalanobis distance and scaled model error is proposed with meaningful stochastic interpretation. The χ^2 distribution interpretation provides theoretical foundation and guideline for the calibration of the main parameters. Last but not least, the nonlinear system identification with linear local models is developed and applied to the turbo-charged engine modeling. Promising results are demonstrated on vehicle field test data. The proposed algorithm requires no prior system knowledge and very little calibration efforts.

2.3 Problem Statement

Composite local model system identification approaches exploit multiple local models to approximate system nonlinearities where each model has certain valid operating regime. The global model is composed of the local models with an interpolation scheme. This divide-and-conquer scheme is similar to the linearization of nonlinear models at different operating points.

Formally, the nonlinear system we want to identify is represented as follows:

$$y(k) = F(x(k); \Theta, Z) = \sum_{i=1}^K \gamma_i(x(k); \zeta_i) f_i(x(k); \theta_i), \quad (2.1)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the system states at time instant k , $f_i(\cdot)$ is the i th local model that can take the form of a point, linear model, Markov chain, neural network, etc. Each model $f_i(\cdot)$ is parameterized by θ_i . For example, θ_i represents the probability transition matrix if $f_i(\cdot)$ is represented as a Markov chain and θ_i represents the regression coefficients and biases if $f_i(\cdot)$ is a linear model. K is the number of local models and $\gamma_i(\cdot)$ is a weighting function that interpolates the local models and is parameterized by ζ_i . $\Theta = [\theta_1, \dots, \theta_K]$ is the collection of all local model parameters and $Z = [\zeta_1, \dots, \zeta_K]$ is the collection of all parameters in the interpolating functions. Note that the composite local model (2.1) includes the Takagi-

Sugeno fuzzy system as a special case where $\gamma_i(x) = \frac{\mu_i(x)}{\sum_{j=1}^K \mu_j(x)}$ with $\mu_j(x)$ being the fuzzy membership function and $f_i(x; \theta_i) = A_i x + b_i$ with A_i and b_i being the parameters.

Given a series of input-output data pairs $\{x(k), y(k)\}_{k=1}^N$, the tasks of identifying the composite local model (2.1) involve the following:

1. Determine the type of model $f_i(\cdot)$ for all local models;
2. Design the interpolation function $\gamma_i(\cdot)$;
3. Decide the number of local models K ;
4. Identify parameters Θ and Z .

While Task 1) is important and interesting, we later consider local linear models due to its practical usefulness and simplicity in control synthesis. In the following sections, we focus on Task 2)- Task 4) where we develop an evolving STF to obtain the interpolation functions; we develop efficient algorithms to update parameters Θ and Z online; and we determine the number of local models K .

Before we describe the main results, we introduce the following definition and lemma that will be used in later developments.

Definition 2.3.1. *Let $Z_1, \dots, Z_k$ be independent, standard normal random variables. The following sum of their squares,*

$$Q = \sum_{i=1}^k Z_i^2, \quad (2.2)$$

is distributed according to the chi-square (χ^2) distribution with k degrees of freedom, i.e., $Q \sim \chi_k^2$.

Lemma 2.3.2. *([29]): Let A, B, C, D be matrices of appropriate dimensions. Suppose matrices A, C , and $C^{-1} + DA^{-1}B$ are nonsingular, then*

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}. \quad (2.3)$$

2.4 Spatial-Temporal Filtering for Local Model Interpolation

In this section, we introduce a STF scheme to interpolate local models. The STF uses ellipsoidal clusters as function bases. Specifically, we associate each local model $f_i(\cdot)$ with a cluster $\mathcal{C}_i = \{v_i, \Sigma_i\}$, $i = 1, 2, \dots, K$ with v_i and Σ_i being, respectively, the cluster centroid and covariance matrix, and K being the number of clusters. Each cluster defines a validity zone for its corresponding local interpolation model. The whole input/output space is separated into subspaces on which a local model is defined and updated.

In order to obtain clusters that characterize the input-output relation, our clustering algorithm considers the extended input-output vector space. When a new input-output data pair $(x(k), y(k))$ comes in, we create the input-output vector $x_a(k) = [x(k), y(k)]$ and the distance between the vector $x_a(k)$ and cluster $\mathcal{C}_i$ is computed using the Mahalanobis distance,

$$\mathcal{M}(x_a(k), \mathcal{C}_i) = \sqrt{(x_a(k) - v_i)^\top \Sigma_i^{-1} (x_a(k) - v_i)}. \quad (2.4)$$

For each local model $f_i(x; \theta_i)$ with θ_i being the local model parameter vector, the model residual can be computed from the input-output data pair $(x(k), y(k))$ and the model parameter at time step k as,

$$e_i(k) = y(k) - f_i(x(k); \theta_i(k)). \quad (2.5)$$

We denote Λ_i , $i = 1, \dots, K$, the residual covariance matrix of local model i . In practice, Λ_i is typically not available as *a priori*. A possible choice is $\Lambda_i = I_{n_y}$, $\forall i = 1, \dots, K$; n_y is the dimension of system output, that is, the residuals are not weighted. Alternatively, if we have a data batch of size N and we run our clustering algorithm described later on, an estimate of Λ_i , $\hat{\Lambda}_i$, can be computed as,

$$\hat{\Lambda}_i = \sum_{k=1}^N \frac{1}{N-1} e_i(k) e_i(k)^\top. \quad (2.6)$$

When the error covariance needs to be estimated online, the following recursive update can be used:

$$\hat{\Lambda}_i(k+1) = (1 - \alpha) \cdot \hat{\Lambda}_i(k) + \alpha \cdot e_i(k) e_i(k)^\top, \quad (2.7)$$

with $\alpha \in (0, 1)$ being the update rate. This update is known as the exponential moving average that assigns more weights to recent data [30], [31]. It is advantageous for applications that emphasize recent data points.

Now let's define a set, $\mathcal{F}_i$, as,

$$\mathcal{F}_i = \left\{ (x, y) | x_a \sim \mathcal{N}(v_i, \Sigma_i), \text{ and } y = f_i(x; \theta_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \Lambda_i) \right\}, \quad (2.8)$$

where $x_a = [x; y]$ is the extended input-output vector. Intuitively, $\mathcal{F}_i$ represents the set of input-output data pair such that the input-output vector x_a is normally distributed with mean v_i and covariance Σ_i , and the residual of local model i is normally distributed with zero mean and covariance Λ_i . The dissimilarity metric between the data point $(x(k), y(k))$ and the set $\mathcal{F}_i$ is then defined as,

$$\begin{aligned} \mathcal{D}((x(k), y(k)), \mathcal{F}_i) &:= \frac{1}{2} \mathcal{M}^2(x_a(k), \mathcal{C}_i) + \frac{1}{2} e_i(k)^\top \Lambda_i^{-1} e_i(k) \\ &= \frac{1}{2} (x_a(k) - v_i)^\top \Sigma_i^{-1} (x_a(k) - v_i) \\ &\quad + \frac{1}{2} (y(k) - f_i(x(k); \theta_i))^\top \Lambda_i^{-1} (y(k) - f_i(x(k); \theta_i)). \end{aligned} \quad (2.9)$$

The dissimilarity metric defined in (2.9) combines the Mahalanobis distance, which characterizes the distance between the point and the cluster, and a scaled model error that characterizes how the new data matches the current local model. The following proposition provides further insights on the proposed dissimilarity metric.

Proposition 2.4.1. *Assume the multivariate normal distribution defined by the cluster (mean and covariance) and its associated local model error distribution are independent. Then for $(X, Y) \in \mathcal{F}_i$, the proposed dissimilarity metric in (2.9) multiplied by 2 is chi-square distributed with degree of freedom $n_x + 2n_y$.*

Proof. Let $(X, Y) \in \mathcal{F}_i$ be a pair of random variables and $X_a = [X; Y]$. Define

$$Z = \Sigma_i^{-\frac{1}{2}} (X_a - v_i) = [Z_1, Z_2, \dots, Z_{n_x+n_y}], \quad (2.10)$$

$$Q = \Lambda_i^{-\frac{1}{2}}(Y - f_i(X; \theta_i)) = [Q_1, Q_2, \dots, Q_{n_y}]. \quad (2.11)$$

It is straightforward to show that $Z \sim \mathcal{N}(\mathbf{0}_{n_x+n_y}, \mathbf{I}_{n_x+n_y})$ and $Q \sim \mathcal{N}(\mathbf{0}_{n_y}, \mathbf{I}_{n_y})$, where n_x and n_y are the dimensions of the input and output vectors, respectively. As a result, the dissimilarity metric between the input-output pair (X, Y) and the set $\mathcal{F}_i$ defined in (2.9) is:

$$\begin{aligned} 2 \cdot \mathcal{D}((X, Y), \mathcal{F}_i) &= (X_a - v_i)^\top \Sigma_i^{-1} (X_a - v_i) + (Y - f_i(X; \theta_i))^\top \Lambda_i^{-1} (Y - f_i(X; \theta_i)) \\ &= Z_1^2 + \dots + Z_{n_x+n_y}^2 + Q_1^2 + \dots + Q_{n_y}^2. \end{aligned} \quad (2.12)$$

With the assumption that the cluster distribution and the error distribution are independent, from Definition 2.3.1 it follows that $2 \cdot \mathcal{D}((X, Y), \mathcal{F}_i)$ is chi-square distributed with degrees of freedom $n_x + 2n_y$, i.e., $2 \cdot \mathcal{D}((X, Y), \mathcal{F}_i) \sim \chi_{n_x+2n_y}^2$. $\square$

Now define the unnormalized weights of data point $(x(k), y(k))$ to set $\mathcal{F}_i$, $\mu_i(k), i = 1, \dots, K$ as,

$$\mu_i(k) = \exp \left\{ -\mathcal{D}((x(k), y(k)), \mathcal{F}_i) \right\}, \quad (2.13)$$

whose stochastic interpretation is given in the following Lemma.

Lemma 2.4.2. *Let $\mathcal{F}_i$ be the set defined in (2.8). The unnormalized weighting function $\mu_i(k)$ defined in (2.13) satisfies:*

$$\mu_i(k) \propto \Pr \left[(x(k), y(k)) \in \mathcal{F}_i \right]; \quad (2.14)$$

Proof. From the definition in (2.8), we have,

$$\begin{aligned} \Pr[(x(k), y(k)) \in \mathcal{F}_i] &= \Pr[x_a(k) \sim \mathcal{N}(v_i, \Sigma_i), y(k) = f_i(x(k); \theta_i) + \epsilon_i] \\ &= \Pr[x_a(k) \sim \mathcal{N}(v_i, \Sigma_i)] \cdot \Pr[y(k) = f_i(x(k); \theta_i) + \epsilon_i | x_a(k) \sim \mathcal{N}(v_i, \Sigma_i)]. \end{aligned} \quad (2.15)$$

$\square$

Note that

$$\Pr[x_a(k) \sim \mathcal{N}(v_i, \Sigma_i)] \propto \exp \left\{ -\frac{1}{2} (x_a(k) - v_i)^\top \Sigma_i^{-1} (x_a(k) - v_i) \right\} \quad (2.16)$$

and from the assumption that the residual $e_i(k)$ is Gaussian distributed with zero mean and covariance Λ_i , we have,

$$\Pr[y(k)|x(k), x_a(k) \sim \mathcal{N}(v_i, \Sigma_i)] \propto \exp\left\{-\frac{1}{2}(y(k) - f_i(x(k); \theta_i))^\top \Lambda_i^{-1}(y(k) - f_i(x(k); \theta_i))\right\} \quad (2.17)$$

Combining Equations (2.15) $\sim$ (2.17), we have,

$$\begin{aligned} \Pr[(x(k), y(k)) \in \mathcal{F}_i] &\propto \exp\left\{-\frac{1}{2}(x_a(k) - v_i)^\top \Lambda_i^{-1}(x_a(k) - v_i) - \right. \\ &\quad \left. \frac{1}{2}(y(k) - f_i(x(k); \theta_i))^\top \Lambda_i^{-1}(y(k) - f_i(x(k); \theta_i))\right\} \\ &\propto \exp\{-\mathcal{D}((x(k), y(k)), \mathcal{F}_i)\} = \mu_i(k), \end{aligned} \quad (2.18)$$

where the equation in the last step comes from the definition in (2.13). This completes the proof.

It is worth noting that Lemma 2.4.2 bridges the proposed dissimilarity (2.9) to the probability of belonging to the cluster/model combo set, which further justifies the dissimilarity metric. The model interpolating weights can be obtained by normalizing $\mu_i(k)$ which resembles the softmax-like function:

$$\begin{aligned} \gamma_i(k) &= \frac{\mu_i(k)}{\sum_{j=1}^K \mu_j(k)} \\ &= \frac{\exp\{-\mathcal{D}((x(k), y(k)), \mathcal{F}_i)\}}{\sum_{j=1}^K \exp\{-\mathcal{D}((x(k), y(k)), \mathcal{F}_j)\}}. \end{aligned} \quad (2.19)$$

Remark 2.4.3. Using (2.9), (2.13), and (2.19) presents a process that transforms the time-series data $(x(k), y(k))$ to a vector of weights that interpolate the local models. On the other hand, as we see later in Algorithm 1, the data points will be used to update the clusters and local models. By virtue of transforming the data vector $(x(k), y(k))$ to the model interpolating weights and the fact that the filtering parameters (cluster features and local model parameters) are evolving over time, we call this process *spatial-temporal filtering*.

With the above dissimilarity and interpolating weights, the proposed evolving clustering algorithm recursively updates the cluster features from the data stream as described in the

first section of Algorithm 1. When a new data pair $(x(k), y(k))$ comes in, if there is no existing cluster, we first create and initialize a cluster with $x_a = [x(k); y(k)]$ as the centroid and an identity matrix $\Sigma_1^{-1} = \beta I_{n_x+n_y}$ as the inverse of the covariance matrix, where β is an initialization parameter. As shown later, the recursive computation of the inverse of the covariance matrix instead of the covariance matrix itself is more computationally efficient. For this first cluster, we also initialize the model parameters as zeros, i.e., $\theta_1 = \mathbf{0}$.

For all existing clusters, we compute the dissimilarities of $(x(k), y(k))$ from all existing clusters using (2.9). We then find the cluster i^* with minimum dissimilarity metric, i.e., $i^* \leftarrow \arg \min_i \mathcal{D}((x(k), y(k)), \mathcal{F}_i)$. If the minimum dissimilarity is greater than the threshold parameter $\mathcal{D}_0$, then we create a new cluster ($K \leftarrow K + 1$) with $v_K = [x(k); y(k)]$, $\Sigma_K^{-1} = \beta I_{n_x+n_y}$. The model parameters are also initialized as zeros.

Note that Proposition 2.4.1 casts light on the choice of the dissimilarity threshold $\mathcal{D}_0$ for the creation of new clusters. With the chi-square distribution interpolation, one can use the cumulative distribution function (CDF) to choose the threshold. If the minimum dissimilarity is less than the threshold parameter $\mathcal{D}_0$, then the centroid and covariance matrix of the closest cluster i^* can be updated with $\alpha \in (0, 1)$ as the updating rate,

$$\Sigma_{i^*}(k) = (1 - \alpha)\Sigma_{i^*}(k - 1) + \alpha(x_a(k) - v_{i^*})(x_a(k) - v_{i^*})^\top, \quad (2.20)$$

$$v_{i^*}(k) = (1 - \alpha)v_{i^*}(k - 1) + \alpha x_a(k), \quad (2.21)$$

Algorithm 2.1 Online nonlinear system identification with spatial-temporal filters.

Parameter: Cluster learning rate α , initial residual covariance Λ_0 , maximum number of clusters $K_{\max}$, inverse cluster covariance initialization parameter β , dissimilarity threshold $\mathcal{D}_0$.

Input : $\{x(k), y(k)\}_{k=1}^N$.

Output : $v_i, \Sigma_i^{-1}, \Lambda_i, \theta_i, i = 1, \dots, K$.

initialize $K \leftarrow 0, P \leftarrow$ empty matrix;

for $k = 1 \rightarrow N$ **do**

 /* Evolving clustering */

if $K = 0$ **then**

initialize $v_1 \leftarrow [x(k); y(k)], \Sigma_1 \leftarrow \beta \mathbf{I}_{n_x+n_y}, \theta_1 \leftarrow \mathbf{0}, \vec{\mathcal{D}} \leftarrow \frac{1}{2} y^\top(k) \Lambda^{-1} y(k);$

increment $K \leftarrow K + 1;$

else

initialize $\vec{\mathcal{D}} \leftarrow \mathbf{0}_{K \times 1};$

for $i = 1 \rightarrow K$ **do**

compute $\mathcal{D}((x(k), y(k)), \mathcal{F}_i)$ using (2.4), (2.5), and (2.9);

assign $\vec{\mathcal{D}}[i] \leftarrow \mathcal{D}((x(k), y(k)), \mathcal{F}_i);$

end

find $i^* \leftarrow \operatorname{argmin}_i \vec{\mathcal{D}}[i], \mathcal{D}_{\min} \leftarrow \min \vec{\mathcal{D}};$

if $\mathcal{D}_{\min} > \mathcal{D}_0$ && $K < K_{\max}$ **then**

increment $K \leftarrow K + 1;$

initialize $v_K \leftarrow [x(k); y(k)], \Sigma_K \leftarrow \beta \mathbf{I}_{n_x+n_y}, \theta_K \leftarrow \mathbf{0};$

augment $\vec{\mathcal{D}} \leftarrow [\vec{\mathcal{D}}; \frac{1}{2} y^\top(k) \Lambda^{-1} y(k)];$

else

update $\Sigma_{i^*}^{-1}$ using (2.22);

update v_{i^*} using (2.21);

update Λ_{i^*} using (2.7);

end

end

compute $\gamma_i(k), i = 1, \dots, K$ using (2.19);

 /* Model parameter update */

update model parameters Θ

end

Note that we need the inverse of Σ_i , $i = 1, \dots, K$ to compute the dissimilarity, so instead of using (2.20) to update Σ_{i^*} and then compute its inversion, we exploit Lemma 2.3.2 to recursively compute the inverse to facilitate the computation with $A = (1 - \alpha)\Sigma_{i^*}(k - 1)$, $B = x_a(k) - v_{i^*}$, $C = \alpha\mathbf{I}_{n_x}$, and $D = (x_a(k) - v_{i^*})^\top$:

$$\begin{aligned}\Sigma_{i^*}^{-1}(k) &= [(1 - \alpha)\Sigma_{i^*}(k - 1) + \alpha(x_a(k) - v_{i^*})(x_a(k) - v_{i^*})^\top]^{-1} \\ &= \frac{1}{1 - \alpha}\Sigma_{i^*}^{-1}(k - 1) - \frac{1}{(1 - \alpha)^2}\Sigma_{i^*}^{-1}(k - 1) \times \\ &\quad \times (x_a(k) - v_{i^*})\psi_{i^*}^{-1}(k)(x_a(k) - v_{i^*})^\top\Sigma_{i^*}^{-1}(k - 1),\end{aligned}\tag{2.22}$$

where $\psi_{i^*}(k) = \frac{1}{\alpha} + \frac{1}{1 - \alpha}(x_a(k) - v_{i^*})^\top\Sigma_{i^*}^{-1}(k - 1)(x_a(k) - v_{i^*})$ is a scalar. By using (2.22), only a scalar inversion is needed. With the updated cluster, the interpolation weights, $\gamma_i(k)$, $i = 1, \dots, K$, are computed using (2.19), which is used to update model parameters. The model parameters update depends on the local model types. We next present the parameter update algorithm for linear local models.

2.5 Online Identification of Composite Linear Local Models

2.5.1 System Reformulation

In Section 2.4, we introduced STFs that use evolving clusters as basis functions to compute the interpolation weights and a nonlinear system identification algorithm with a general local model is presented. Since practical control designs are predominately based on linear systems, in this section, we treat the local models of linear form as follows:

$$\begin{aligned}y(k) &= a_i^{(1)}y(k - 1) + a_i^{(2)}y(k - 2) + \dots + a_i^{(d_y)}y(k - d_y) \\ &\quad + b_i^{(1)}u(k - 1) + b_i^{(2)}u(k - 2) + \dots + b_i^{(d_u)}u(k - d_u) + c_i,\end{aligned}\tag{2.23}$$

where d_u and d_y are again the input delay and output delay, respectively; $a_i^{(1)}, \dots, a_i^{(d_y)}, b_i^{(1)}, b_i^{(d_u)}, c_i$ are system matrices to be determined for local model i . Define $A_i = [a_i^{(1)}, \dots, a_i^{(d_y)}, b_i^{(1)}, \dots, b_i^{(d_u)}]$ and $x(k) = [y(k - 1); \dots; y(k - d_y); u(k - 1); \dots; u(k - d_u)]$. Then (2.23) can be written as:

$$y(k) = A_i x(k) + c_i.\tag{2.24}$$

The nonlinear model can now be represented by the following composite model:

$$y^{(j)}(k) = \sum_{i=1}^K \gamma_i A_i^{(j)} x(k) + \sum_{i=1}^K \gamma_i c_i^{(j)}, \quad j = 1, \dots, n_y, \quad (2.25)$$

where the superscript j represents the j th row. Instead of estimating matrices A_i , c_i as an entire matrix, we prefer to recursively identify each row of these matrices. Let $\theta^{(j)} = [A_1^{(j)}, c_1^{(j)}, \dots, A_K^{(j)}, c_K^{(j)}]^\top$ and $\phi^\top(k) = [\gamma_1(k)x^\top(k), \gamma_1(k), \dots, \gamma_K(k)x^\top(k), \gamma_K(k)]$, then (2.25) becomes,

$$y^{(j)}(k) = \phi^\top(k) \theta^{(j)}. \quad (2.26)$$

Given a batch of dataset $\{x(k), y(k)\}_{k=1}^N$, the system (2.26) can be augmented as

$$Y_N^{(j)} = \Phi_N^\top \theta^{(j)}, \quad (2.27)$$

where $Y_N^{(j)} = [y^{(j)}(1), \dots, y^{(j)}(N)]^\top$ and $\Phi_N = [\phi(1) \dots \phi(N)]$. The least square estimate of $\theta^{(j)}$, $\hat{\theta}^{(j)}$, is

$$\hat{\theta}^{(j)}(N) = [\Phi_N \Phi_N^\top]^{-1} \Phi_N Y_N^{(j)}. \quad (2.28)$$

2.5.2 Recursive Least Squares

For the proposed method, we estimate the parameters recursively, i.e., update the parameters online with continuously received data pair $(x(k), y(k))$. Towards that end, we exploit the recursive least square (RLS) approach. The following are key RLS parameter update equations:

$$\epsilon^{(j)}(k) = y^{(j)}(k) - \phi^\top(k) \hat{\theta}_{k-1}^{(j)}, \quad (2.29a)$$

$$\hat{\theta}^{(j)}(k) = \hat{\theta}_{k-1}^{(j)} + P_k \phi(k) \epsilon^{(j)}(k), \quad (2.29b)$$

$$P_k = \frac{1}{\lambda} \left(P_{k-1} - \frac{P_{k-1} \phi(k) \phi^\top(k) P_{k-1}}{\lambda + \phi^\top(k) P_{k-1} \phi(k)} \right). \quad (2.29c)$$

Readers can refer to [14] for detailed RLS derivations. With RLS, the following algorithm block can replace Line 24 in Algorithm 1 for a complete nonlinear system identification algorithm.

Algorithm 2.2 RLS model parameter update.

```

form  $\phi \leftarrow [\gamma_1(k)x^\top(k), \gamma_1(k), \dots, \gamma_K(k)x^\top(k), \gamma_K(k)]^\top$ ;
    /* Check if a new cluster is created */
if  $\text{size}(P, 1) < K * (n_x + 1)$  then
    |  $P = \text{diag}\{P, \beta \mathbf{I}_{n_x+1}\}$ ;
end
update  $P$  using (2.29c);
for  $j = 0 \rightarrow n_y$  do
    | form  $\theta^{(j)} \leftarrow [A_1^{(j)}, c_1^{(j)}, \dots, A_K^{(j)}, c_K^{(j)}]^\top$ ;
    | compute the model residual  $\epsilon = y^{(j)}(k) - \phi^\top \theta^{(j)}$ ;
    | update the parameters  $\theta^{(j)}$  using (2.29b);

    /* Unroll the parameter vector into system matrices */
    for  $i = 1 \rightarrow K$  do
    | |  $A_i[j, :] \leftarrow \theta^{(j)}[(i-1) * (n_x + 1) + 1 : i * (n_x + 1) - 1]$ ;
    | |  $c_i[j] \leftarrow \theta^{(j)}[i * (n_x + 1)]$ ;
    | end
end

```

2.5.3 Towards Implementation

Note we use input-output pairs in the training process to define interpolation weights based on cluster features and system matrices. When predicting, only input data x is available. To obtain the interpolating weights, we first use the identified system matrices to predict the output from each local model, i.e., $\hat{y}_i = A_i x + c_i$ and then compute the dissimilarity $\mathcal{D}((x, \hat{y}_i), \mathcal{F}_i)$, $\forall i = 1, \dots, K$. Then the degree of membership is computed as

$$\gamma_i = \frac{\exp \{-\mathcal{D}((x, \hat{y}_i), \mathcal{F}_i)\}}{\sum_{j=1}^K \exp \{-\mathcal{D}((x, \hat{y}_j), \mathcal{F}_j)\}}, \quad (2.30)$$

where we use the local predicted output $\hat{y}_i, i = 1, \dots, K$. Note that due to the nonlinearity of the interpolating weights (2.30), the overall composed model produced by the STF is nonlinear even when the local models are linear.

The proposed online system identification algorithm tracks the cluster covariance using (2.22) and model error covariance using (2.7). These covariance matrices characterize the accuracy of the model and dispersion of the cluster. Therefore, for online prediction, one can set thresholds on these two matrices and only use local models/clusters with small enough covariance. We refer the readers to our conference paper [7] for a numerical example.

2.6 Application to Turbo-Charged Engine System Identification

2.6.1 Simulation-based Validation

In this subsection, we validate the proposed system identification algorithm on a high-fidelity engine simulator from Ford. The inputs we use are engine breathing related signals, i.e. throttle, CAMs, fuel pulse width, fuel injection timing, spark time and wastegate. The output we use is the fuel consumption rate. The training data is generated using the pseudo-random multilevel (PRML) method [32] and the trained model is validated on three standard driving cycles including FTP75, FTP-Highway and NEDC. The data sampling rate is 100 milliseconds.

Table 2.1 Parameters for engine system identification with a high-fidelity engine simulator.

α	Λ_0	$K_{\max}$	β	$\mathcal{D}_0$	λ	d_y	d_u
0.001	I	10	1	12.2421	0.98	1	1

The parameters of the proposed nonlinear online system identification algorithm are listed in Table 2.1. The choices of input delay d_u and output delay d_y are mainly constrained by the memory and computational resources available. Note that increasing d_u by m will increase the dimension of matrices of Σ_i by $m \cdot n_x$, where n_x is the dimension of inputs (7 in the engine system). Therefore, one may want to keep d_u small (e.g., less than 3). Since the dimension of n_y is low, one can choose a larger d_y if the onboard computing resources are sufficient. We would like to keep them low so we choose $d_u = 1$ and $d_y = 1$ to facilitate further real-time vehicle implementation. However, one can run cross-validations to determine the best

combinations of permissible d_u and d_y .

The parameter β used to specify the inverse of the cluster covariance should not be too small, because a small β will generate a wide-range cluster to cover incoming data points, this may include data points that are far away. Similarly, the parameter Λ_i that initializes the error covariance should not be too large, it is better to be set at the same level as cluster covariance. If a larger or smaller initial value is picked for cluster covariance or error covariance, the dissimilarity metric will have high possibility to be dominated by either the scaled error term or the Mahalanobis distance term. So we set $\Lambda_i = I$ and $\beta = 1$ according to the normalized input-output dataset.

The learning rate α is chosen as 0.001 based on 10-fold cross-validation. Large α may make the algorithm diverge but if α is too small, the learning process can be slow. Note that we use $\alpha = 0.001$ other than 0.01 used in the numerical example in our conference paper [7] as the underlying engine system is higher-dimensional and more complex than the sinusoidal function. For the choice of forgetting factor λ , we pick $\lambda = 0.98$ based on our personal experience and 10-fold cross-validation.

The maximum number of models, K_{max} , is generally constrained by the available resources since more local models require more memory and computational resources for implementation. Last but not least, the dissimilarity threshold $\mathcal{D}_0$ controls when to start a new model and the chi-square distribution interpretation by Proposition 2.4.1 provides a good reference range.

We compare the performance with a well-trained neural network model from a Ford supplier. The network has a single hidden layer with ten neurons. The performance comparison is shown in Figure 2.2, where the performance metric *Best Fit Rate* (BFR) is defined as $BFR = \max \left\{ 1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}, 0 \right\} * 100\%$. Here y is the vector of true output; $\hat{y}$ is the vector of estimated output; and $\bar{y}$ is the mean of the true output. Six local models are generated with our STF method and the performance of the STF method is superior to the neural network model as shown in Fig. 2.2.

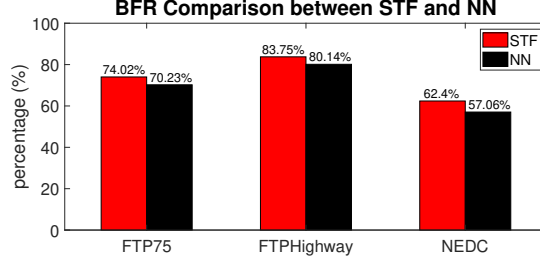


Figure 2.2 Performance comparison between STF and Neural Network model (one-step prediction).

2.6.2 Experimental Vehicle Data Validation

In this subsection, we present results of application of the proposed online nonlinear system identification to the turbo-charged engine system described in Section II. The system has seven inputs and we use fuel consumption as the output. The data is collected in a field test drive from an experimental vehicle with a 3.5L turbo-charged prototype engine at 50 millisecond sampling rate.

With the input-output data stream, clusters and local models are generated according to Algorithms 1 and 2. To examine the physical interpretations of the models, we investigate the model generation timings. One example is shown in Fig. 2.3, which corresponds to a transient model generation where the ECAM aggressively advances at open throttle and retarded ICAM. Similar patterns are found in other model generation timings but we only show Fig. 2.3 due to space limitations. This analysis shows that the local models can characterize reasonable engine operation patterns.

Through the training on the experimental vehicle data, six clusters/models are generated. To validate the model performance, we first run one-step prediction on another set of experimental vehicle data that is not used in training. As shown in Fig. 2.4, the prediction gives a BFR rate of 83.39% and the model can capture the transient dynamics reasonably well as depicted in the zoomed-in window. The contribution of each signal in the identified linear models is shown in Fig. 2.5. The coefficients of $y(k-1)$ term in the local models are around 0.8, which naturally shows its big impact on the predicted $y(k)$. On the other

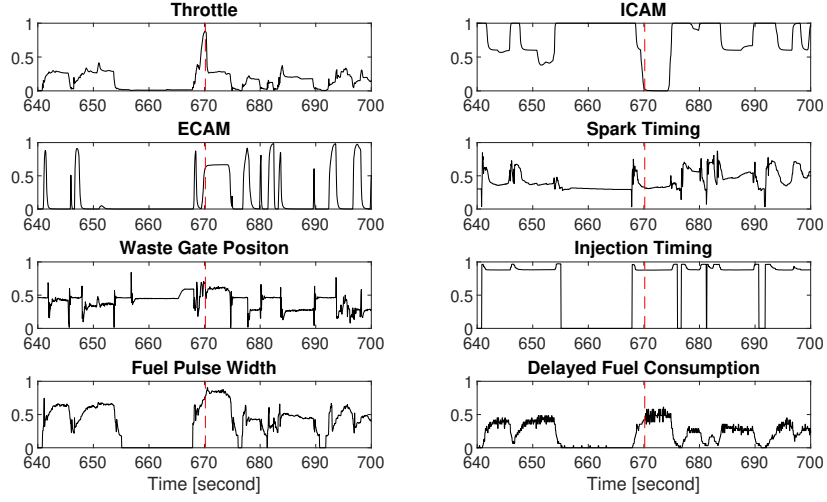


Figure 2.3 One model generation timing with aggressive ECAM advancing at open throttle and retarded ICAM.

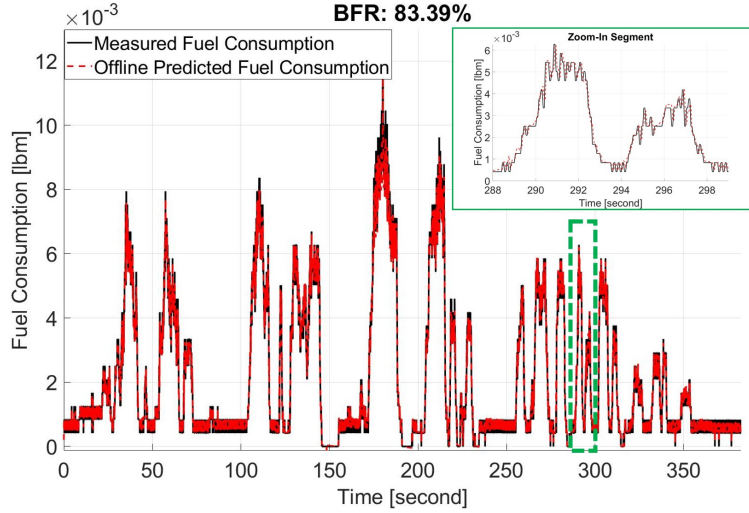


Figure 2.4 One-step fuel rate prediction on validation data.

hand, the input signals u also play an important role as can be seen from the weights. Note that the validation data may contain driving scenarios that are not covered in the training dataset. Therefore, we believe the performance can be improved with more comprehensive driving dataset.

As one of the main motivations of identifying the models is for predictive control, we further evaluate the multi-step as well as the open-loop simulation performance of the identified

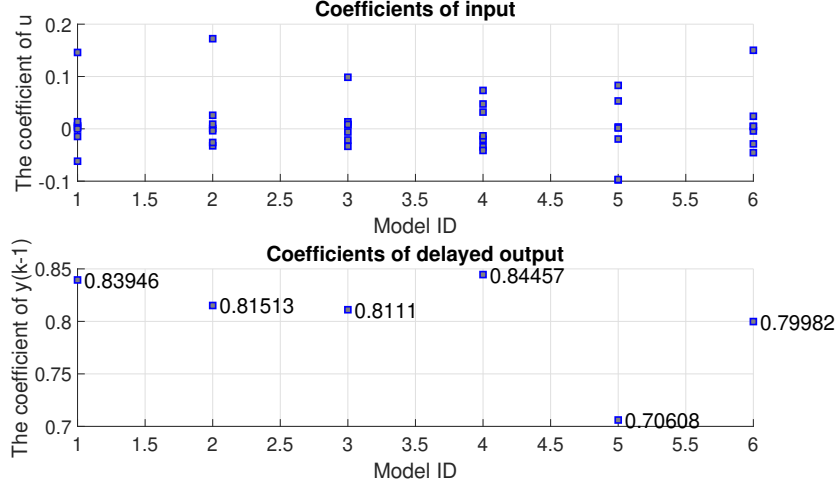


Figure 2.5 Coefficients of sub-models.

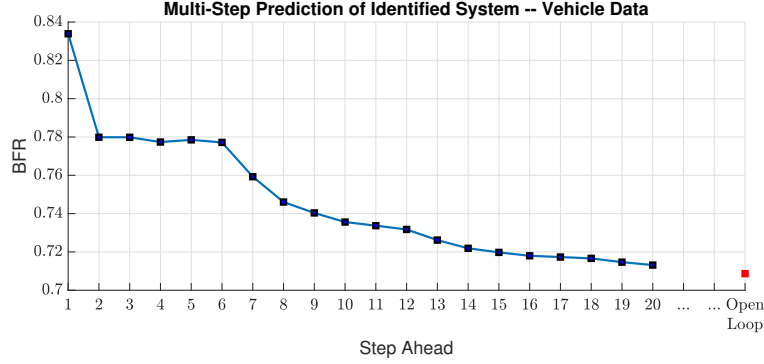


Figure 2.6 Multi-step prediction performance.

models. The multi-step and open-loop predictions are both performed by using *predicted* outputs from earlier steps along with the control inputs for prediction. The performance from one-step to twenty-step predictions are illustrated in Fig. 2.6. Not surprisingly, the prediction performance decreases as the prediction horizon increases. However, the open-loop simulation can still offer a BFR of 70.87%. In our future work, we will investigate techniques to improve the open-loop prediction performance, including enhancing the training dataset with improved design of experiments (DOE) and cultivating the RLS to incorporate loss function for open-loop prediction.

We next investigate the algorithmic robustness by feeding the same training data to the

algorithm multiple times while using the results from last round as the initial model for the next round. The algorithmic robustness can then be characterized by the fluctuations of the parameters. Towards that end, we define the parameter fluctuation index (PFI) as $PFI = ||params(k+1) - params(k)||$, where $params(k)$ represents the parameter vector at time step k . The PFI s for the local models are all within a very small range $0 \sim 10^{-5}$, showing the robustness of the proposed algorithm.

The proposed system identification algorithm was embedded in the experimental vehicle and the execution time is ~ 10 ms with $K_{\max} = 10$ in the engine PCM, which shows the efficiency of the algorithm and the feasibility for online implementation. We also used the identified model in a nonlinear model predictive control (MPC) and showed 10% reduction in fuel consumption with similar speed tracking error as compared to the Ford baseline controller, which demonstrates the effectiveness of the identified model for control purposes. Details about the MPC implementation will be reported in our future publications.

2.7 Conclusion

In this chapter, we introduced an online nonlinear system identification methodology with a composite local model structure for turbo-charged engines. A novel spatial-temporal filtering method is designed by using evolving clusters as function bases to interpolate local models. The algorithm works in an online fashion where it simultaneously updates the clusters and the local models with a training data stream. We applied the online system identification algorithm to a turbo-charged engine system with seven inputs and two outputs. Promising results were demonstrated in experimental test data. Although linear local models were used for the engine application, the proposed method has a versatile structure that can work with arbitrary combinations of local models for other applications, which we intend to investigate in the future. We will also extend our clustering method such that the clusters/models can merge and split based on the covariances and Kullback–Leibler divergence. We will investigate the MPC designs based on the identified model structure.

CHAPTER 3

STOCHASTIC MODEL PREDICTIVE CONTROL FOR QUASI LINEAR PARAMETER VARYING SYSTEMS

3.1 Introduction

Since the last decades, driven by fast technological progress in industrial community, advanced control strategies have been in demand. Model Predictive Control (MPC) [33] is one of the most popular methods that get a lot of attention and widely spread. However, with the development of more and more sophisticated system, nonlinear embedding of MPC design is imperative. For a standard nonlinear MPC (NMPC), it is actually limited in the industrial application due to its nonlinear programming nature, which is intractable for real-time optimization, even if it could lead to very elegant solutions.

To alleviate the problem, linear parameter varying (LPV) model [34] emerges as an alternative to represent the nonlinear systems and can be used for efficient MPC online application. For a LPV model, the nonlinearities are embedded by parameterized linear systems with time-varying scheduling variables. Therefore, the LPV model preserves some advantageous properties of linear time-invariant system, facilitating controller synthesis. In general, LPV systems can be classified into two categories: 1) general LPV structure, for which the scheduling variable is exogenous (i.e., independent from system state and control input), and 2) quasi LPV (qLPV) structure with the scheduling variable being endogenous (i.e., state- and/or control- dependent).

For the general LPV structure, the scheduling is only measurable at the current time instant. Hence, the main difficulty of a general LPV-MPC is to deal with the uncertainty of the scheduling variables during the prediction horizon. The common methods usually fall to robust MPC (RMPC) scheme with the assumption about the boundedness of the scheduling variables, i.e., variation within a convex polytope [35]–[37]. Less conservative

solutions could be achieved by considering the parameter variation rate [38]–[40]. Another way to handle such uncertainty is given by stochastic MPC (SMPC) formulation where the scheduling variables and disturbances are modeled as stochastic processes [41], [42]. The SMPC framework has been developed for linear systems with additive and multiplicative uncertainties, including stochastic tube MPC [43]–[46] and scenario-based MPC [47]–[49], which have later been extended to the general LPV framework [50], [51].

Different from the general LPV structure, the scheduling variable under the qLPV form is a function (possibly nonlinear) of system state and/or control input. Because of the endogenous scheduling variable, the MPC design for the qLPV model often requires an on-line solution to a nonlinear optimization problem, resulting in high computation burden. Several scheduling variable estimation schemes have been proposed to address this problem. [52], [53] utilize the prediction system states in last step to compute the current scheduling variable sequence for MPC optimization. In [54], the linear autoregressive model is combined with least square algorithm to approximate the scheduling variable. These strategies turn the nonlinear optimization problem into a quadratic programming (QP) one, which is more tractable for on-line calculation. So far, the qLPV form based MPC has been applied in many areas such as hydraulic process in chemical industry [55], robotic manipulator set-point tracking and obstacle avoidance [56], and semi-active suspension control for passenger comfort [57]. Note that all these works do not consider the system disturbance related to the modeling error, estimation error, or external noise.

In this chapter, the qLPV form based SMPC for discrete-time nonlinear systems with stochastic disturbances is studied. By leveraging our prior researches [7], [8], the unknown nonlinear dynamics is estimated with evolving spatial-temporal filters (eSTF) whose results is naturally in the qLPV form. We consider chance constraints and exploit the concept of probabilistic reachable set (PRS) [58] to transform the probabilistic constraints of the qLPV system to deterministic ones for robust constraint satisfaction. In addition, similar to the robust tube MPC [59], the actual system is decomposed into a nominal system and an

error dynamic system to facilitate the MPC formulation. The chance constraints imposed on the actual system is transferred to a deterministic ones on the nominal system via PRS, which is functioned as the probabilistic tube to confine the error dynamics with a specified probability level. After tightening the constraints, a SMPC problem is formulated, and the recursive feasibility is guaranteed. To ensure the computation tractability of the SMPC problem, a shifted scheduling variable method is introduced, which exploits the historical prediction state and input to determine the scheduling variable sequence in the current step. In summary, the contributions of this research are:

- A comprehensive formulation of the qLPV-SMPC scheme for treating general nonlinear systems subject to additive disturbance and chance constraints, from model construction, problem formulation, and algorithm implementation.
- The extension of the probabilistic reachable set-based constraint tightening approach to deal with stochastic disturbance and chance constraints for qLPV systems with less conservativeness as compared to [51], [60].
- The development of a shifted scheduling variable approach to efficiently solve the qLPV-SMPC problem as a series of QP problems, facilitating its implementation in real-time applications.
- A case study on the automotive engine control problem to demonstrate the promising performance and computation efficiency in a pseudo-plant generated from real-world vehicle data.

This chapter is organized as follows: In Section 3.2, the qLPV model and the considered optimal control problem are introduced. The constraint tightening for qLPV system and the corresponding MPC formulation are described in Section 3.3. The reformulation of the SMPC optimization problem into a QP form is discussed in Section 3.4. Section 3.5 demonstrates the qLPV-SMPC method by a case study on an automotive engine model. The conclusion and future work are summarized in Section 3.6.

Nomenclature $\mathbb{N}/\mathbb{N}_+$ is the set of non-negative/positive integers. Given two sets $\mathcal{X}$ and $\mathcal{W}$ of $\mathbb{R}^n$, the notation $\mathcal{X} \ominus \mathcal{W} \triangleq \{x \in \mathcal{X} \mid x + w \in \mathcal{X} \ \forall w \in \mathcal{W}\}$ refers to the Pontrygin set difference. The Gaussian distribution of a random variable x is specified as $x \sim \mathcal{N}$. The expected value and variance of a random variable x are $\mathbb{E}(x)$ and $\text{var}(x)$, respectively, with system information (if known) up to current time instant. The probability and conditional probability are denoted by $\Pr(A)$ and $\Pr(A|B)$. The weighted 2-norm is $\|x\|_P \triangleq \sqrt{x^\top P x}$, where $P \succ (\succcurlyeq) 0$ means a positive (semi-)definite matrix. We refer to quantities of the system realized in closed-loop at time instant k using parentheses, e.g., $x(k)$ is the state measured at time instant k , while quantities used in the MPC prediction are indexed with subscript, e.g., $x_{i|k}$ is the predicted system state i time-steps ahead of current time instant k . $\text{Conv}\{\cdot\}$ is the convex hull of the sets included. The symbol $\otimes$ refers to kronecker product and $\mathbf{1}_N$ is a column vector with ‘1’ in all N rows.

3.2 Problem Formulation

Consider a nonlinear system that can be represented by the following discrete-time nonlinear dynamics:

$$x(k+1) = f(x(k), u(k), \delta(k)), \quad (3.1)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state vector; $u(k) \in \mathbb{R}^{n_u}$ is the control input vector; $\delta(k) \in \mathbb{R}^{n_x}$ is the disturbance vector; and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ represents the nonlinear function that describes the underlying dynamics. To ensure safe and efficient operations under stochastic disturbance $\delta(k)$, the plant is subject to a set of chance constraints on both states and inputs, which are described by

$$\Pr(x(k) \in \mathcal{X}^l \mid x(0)) \geq p_x^l, \ l = 1, \dots, c_x, \forall k \in \mathbb{N}_+, \quad (3.2a)$$

$$\Pr(u(k) \in \mathcal{U}^l \mid x(0)) \geq p_u^l, \ l = 1, \dots, c_u, \forall k \in \mathbb{N}_+, \quad (3.2b)$$

where $\mathcal{X}_l \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U}_l \subseteq \mathbb{R}^{n_u}$ represent the convex sets with respect to the l th state constraint and input constraint, respectively; and p_x^l and p_u^l are the corresponding prescribed

probability levels of constraint satisfaction. These probability parameters specify the tolerance levels of constraint violations; higher probability levels correspond to harder constraints whereas soft constraints can be specified with lower probability levels.

MPC is known as an effective tool to control dynamic systems with constraints, and the aim of this chapter is to develop an MPC scheme for the general nonlinear plant (3.1). Many engineering systems are inherently nonlinear, and their physical models are not always available. In addition, the nonlinear dynamics leads to a nonlinear MPC (NMPC) problem that involves solving non-convex optimization problems online, the complexity of which hinders its real-time implementations. Based on the above observations, we exploit the evolving spatial-temporal filtering (eSTF)-based system identification method developed introduced in Chapter 2 to approximate the unknown nonlinear dynamics (3.1) into a qLPV form. Specifically, given the input-output data of the nonlinear plant, the proposed identification method uses evolving clusters as function bases to interpolate local linear models with nonlinear composition weights. The nonlinear plant can then be represented as:

$$x(k+1) = \sum_{j=1}^{n_\gamma} \gamma_j(k) (A_j x(k) + B_j u(k)) + w(k), \quad (3.3)$$

where the pair (A_j, B_j) represents the j th local model, $\gamma_j(k) = h_j(x(k), u(k)) \in (0, 1)$ is the nonlinear, softmax-based weighting function, depending on system state and control input, and $\sum_{j=1}^{n_\gamma} \gamma_j(k) = 1$ with n_γ being the number of local models. The additive disturbance $w(k) \in \mathbb{R}^{n_x}$, lumping process disturbance $\delta(k)$ and modeling mismatch, is assumed to be independent and identically distributed (i.i.d.) with zero mean and known covariance Σ_w . By defining $A(\gamma(k)) = \sum_{j=1}^{n_\gamma} \gamma_j(k) A_j$ and $B(\gamma(k)) = \sum_{j=1}^{n_\gamma} \gamma_j(k) B_j$, (3.3) can be written into a discrete-time LPV form:

$$x(k+1) = A(\gamma(k))x(k) + B(\gamma(k))u(k) + w(k), \quad (3.4)$$

where the function $\gamma(k)$ is the scheduling variable belonging to the following unit simplex:

$$\Lambda \triangleq \left\{ \gamma \in \mathbb{R}^{n_\gamma} \mid \sum_{j=1}^{n_\gamma} \gamma_j = 1, \gamma_j \in (0, 1] \right\}. \quad (3.5)$$

As a result, the system matrices $A(\gamma)$ and $B(\gamma)$ are members of the vertex-based polytopic matrix family

$$\Omega \triangleq \left\{ (A(\gamma), B(\gamma)) = \sum_{j=1}^{n_\gamma} \gamma_j (A_j, B_j), \gamma \in \Lambda \right\}. \quad (3.6)$$

After representing the nonlinear plant with a LPV model, the objective now is to design a stochastic MPC to obtain the optimal control sequence at each time k that minimizes the following finite-horizon expected cost

$$J_N = \mathbb{E} \left[\sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}) \right], \quad (3.7)$$

while satisfying the system dynamics (3.4) and chance constraints (3.2). Here N is the optimization horizon, $x_{i|k}$ and $u_{i|k}$ are the system state and control input predicted i steps ahead of current time instant k , and $\ell(x_{i|k}, u_{i|k})$ and $V_f(x_{N|k})$ are, respectively, the stage cost and terminal cost defined as:

$$\ell(x_{i|k}, u_{i|k}) \triangleq \|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2, \quad (3.8a)$$

$$V_f(x_{N|k}) \triangleq \|x_{N|k}\|_P^2, \quad (3.8b)$$

with $Q = Q^\top \succ 0$, $R = R^\top \succ 0$, and $P = P^\top \succ 0$.

3.3 Stochastic MPC for Quasi LPV Model

In this section, we develop a tube-based stochastic model predictive control (SMPC) framework where we decompose the original system (3.4) into a nominal system and an error dynamic system as:

$$x(k) = z(k) + e(k), \quad (3.9a)$$

$$u(k) = Ke(k) + v(k), \quad (3.9b)$$

so that the nominal system state $z(k)$ and error $e(k)$ evolve according to

$$z(k+1) = A(\gamma(k))z(k) + B(\gamma(k))v(k), \quad (3.10a)$$

$$e(k+1) = A_K(\gamma(k))e(k) + w(k), \quad (3.10b)$$

where $v(k)$ is the nominal control input, and $A_K(\gamma(k)) = A(\gamma(k)) + B(\gamma(k))K$ with K being a constant control gain. The nominal system state $z(k)$ is initialized at $z(0) = x(0)$ (i.e., $e(0) = 0$). We assume that the polytopic LPV model (3.10a), subject to (3.5) and (3.6), is quadratically stabilizable [61], i.e., there exists a constant control gain K such that $A_K(\gamma(k)) = A(\gamma(k)) + B(\gamma(k))K$ is stable, for all $\gamma(k) \in \Lambda$. The control gain K is introduced to penalize the deviation error between $x(k)$ and $z(k)$, and can be obtained by solving a linear matrix inequality problem with the inequality set formed simultaneously on each vertex of the polytopic matrix family Ω [35].

3.3.1 Chance Constraint Handling

The probabilistic reachable set (PRS) for the error system (3.10b) is introduced next to achieve constraint tightening such that the chance constraints on x and u can be satisfied through the deterministic constraints on z and v . PRS is the probabilistic analogy of robust reachable sets and has been used in [58] for stochastic MPC of LTI systems. In the following, the definition of PRS for the error system (3.10b) is given firstly, and the computational techniques are then introduced to approximate the PRS of the proposed qLPV system for efficient computations.

Definition 3.3.1 (Probabilistic Reachable Set [58]). *A set $\mathcal{R}$ is a probabilistic reachable set of probability level p for system (3.10b) initialized at $e(0)$ if*

$$\Pr(e(k) \in \mathcal{R} \mid e(0) = 0) \geq p, \quad \forall k \in \mathbb{N}_+. \quad (3.11)$$

Based on Definition 4.2.1, it can be shown that the chance constraints in (3.2) can be converted to the following deterministic constraints on $z(k)$ and $v(k)$:

$$z(k) \in \mathcal{Z} = \bigcap_{l=1}^{c_x} \left(\mathcal{X}^l \ominus \mathcal{R}^{x,l} \right), \quad (3.12a)$$

$$v(k) \in \mathcal{V} = \bigcap_{l=1}^{c_u} \left(\mathcal{U}^l \ominus \mathcal{R}^{u,l} \right), \quad (3.12b)$$

where $\mathcal{Z}$ and $\mathcal{V}$ are the state and control input constraint sets in the nominal system, and $\mathcal{R}^{x,l}$ and $\mathcal{R}^{u,l}$ are the PRS applied on the state error $e(k)$ and the input error $Ke(k)$ corresponding to the l th constraint with probability level p_x^l and p_u^l , respectively.

We now introduce how to calculate $\mathcal{R}^{x,l}$, and the calculation of $\mathcal{R}^{u,l}$ can be performed via similar procedures. We extend the PRS calculations of LTI systems [58], [62] to the considered qLPV system. Specifically, we exploit multiple PRS for LTI models to approximate $\mathcal{R}^{x,l}$ as:

$$\mathcal{R}^{x,l} = \text{Conv}\{\mathcal{R}_j^{x,l}, j \in \{1, \dots, n_\gamma\}\}, \quad (3.13)$$

where $\mathcal{R}_j^{x,l}$ is the PRS for the LTI error system $e(k) = (A_j + B_j K) e(k) + w(k)$ with the pair (A_j, B_j) being the vertex of the polytope Ω . Based on $e(0) = 0$ and the assumption that $w(k)$ is i.i.d with zero mean and known covariance Σ_w , it can be concluded that $\mathbb{E}[e(k+1)] = (A_j + B_j K) \mathbb{E}[e(k)] + \mathbb{E}[w(k)] = 0$. By applying the multivariate Chebyshev inequality [63] and $\mathbb{E}(e(k)) = 0$, the ellipsoidal PRS for the error system $e(k) = (A_j + B_j K) e(k) + w(k)$ with probability level p_x^l is constructed as

$$\mathcal{R}_j^{x,l} = \{e \mid e^\top \Sigma_{j,\infty}^{-1} e < \tilde{p}_x^l\}, \quad (3.14)$$

where $\Sigma_{j,\infty}$ is the solution to the Lyapunov equation $(A_j + B_j K) \Sigma_{j,\infty} (A_j + B_j K)^\top - \Sigma_{j,\infty} = -\Sigma_w$, and $\tilde{p}_x^l = \frac{n_x}{1-p_x^l}$. If $w(k)$ satisfies Gaussian distribution, then $\tilde{p}_x^l = \chi_{n_x}^2(p_x^l)$, where $\chi_{n_x}^2(\cdot)$ is the quantile function of the chi-square distribution with n_x degrees of freedom. The computation of PRS $\mathcal{R}^{u,l}$ for the input constraints can follow the same procedure by simply transferring the dynamics from $e(k)$ to $Ke(k)$.

3.3.2 qLPV-SMPC with Indirect Feedback

The SMPC method relies on predictions over a finite horizon N by using system models (3.4) and (3.10). To differentiate the predicted variables and the actual system variables

with different notations, the prediction model is described by:

$$x_{i+1|k} = A(\gamma_{i|k})x_{i|k} + B(\gamma_{i|k})u_{i|k} + w_{i|k}, \quad (3.15a)$$

$$z_{i+1|k} = A(\gamma_{i|k})z_{i|k} + B(\gamma_{i|k})v_{i|k}, \quad (3.15b)$$

$$e_{i+1|k} = A_K(\gamma_{i|k})e_{i|k} + w_{i|k}, \quad (3.15c)$$

where $w_{i|k}$ follows the same distribution as $w(k+i)$ for $i = 0, \dots, N-1$, and $A_K(\gamma_{i|k}) = A(\gamma_{i|k}) + B(\gamma_{i|k})K$ with $\gamma_{i|k}$ being the predictive scheduling parameter which depends on $z_{i|k}$ and $v_{i|k}$. $x_{0|k}$ and $z_{0|k}$ are selected as $x_{0|k} = x(k)$ and $z_{0|k} = z(k)$, so that $e_{0|k} = x(k) - z(k)$ is generally not zero and contains the feedback information from $x(k)$. Let the time-varying scheduling parameter sequence be denoted by $\mathbf{\Gamma}_k = \{\gamma_{0|k}, \dots, \gamma_{N-1|k}\}$, and the nominal control sequence be $\mathbf{v}_k = \{v_{0|k}, \dots, v_{N-1|k}\}$. With the nominal state initialized at $z(0) = x(0)$, the qLPV-SMPC problem is formulated:

$$\min_{\mathbf{v}_k} J(x_{0|k}, \mathbf{v}_k, \mathbf{\Gamma}_k) = \mathbb{E} \left[\sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}) \right], \quad (3.16a)$$

$$s.t. \quad z_{i+1|k} = A(\gamma_{i|k})z_{i|k} + B(\gamma_{i|k})v_{i|k}, \quad (3.16b)$$

$$e_{i+1|k} = A_K(\gamma_{i|k})e_{i|k} + w_{i|k}, \quad (3.16c)$$

$$x_{i+1|k} = z_{i+1|k} + e_{i+1|k}, \quad (3.16d)$$

$$u_{i|k} = K e_{i|k} + v_{i|k}, \quad (3.16e)$$

$$z_{i|k} \in \mathcal{Z}, \quad (3.16f)$$

$$v_{i|k} \in \mathcal{V}, \quad (3.16g)$$

$$z_{N|k} \in \mathcal{Z}_f, \quad (3.16h)$$

$$x_{0|k} = x(k), \quad z_{0|k} = z(k), \quad e_{0|k} = x_{0|k} - z_{0|k}, \quad (3.16i)$$

for all $i = 0, \dots, N-1$. Note that for $k > 0$, $z(k)$ is taken as $z_{1|k-1}$ rather than resetting it to the actual system state $x(k)$. Therefore, no direct feedback from $x(k)$ is injected into the nominal state $z(k)$. Nevertheless, under the relationship $x(k) = z(k) + e(k)$ (i.e., $x_{0|k} = z_{0|k} + e_{0|k}$), indirect feedback on $z(k)$ is provided through the performance index

(3.16a) of the MPC problem. More details and discussion about the indirect feedback can be found in [62]. Here, $\mathcal{Z}$ and $\mathcal{V}$ are the deterministic nominal system state and control input constraint sets that are obtained according to (3.12), and $\mathcal{Z}_f \subseteq \mathcal{Z}$ is an invariant terminal constraint set that satisfies the following assumption.

Assumption 3.3.2 (Terminal Invariance). *The terminal set $\mathcal{Z}_f \subseteq \mathcal{Z}$ is positively invariant, that is, for all $z \in \mathcal{Z}_f$ and $\gamma \in \Lambda$, there exists $v = Kz \in \mathcal{V}$ such that $A_K(\gamma)z \in \mathcal{Z}_f$.*

The terminal set is introduced to guarantee the recursive feasibility of (3.16). After solving the stochastic optimization control problem (3.16), the first element of the optimal control sequence for nominal system (3.10a) is applied, i.e., $v(k) = v_{0|k}^*$, whereas the optimal input for actual system is applied based on (3.9b). The closed-loop state is then set as $z(k+1) = z_{1|k}$ for the next MPC iteration. The recursive feasibility is next established in the following theorem.

theorem 3.3.3 (Recursive Feasibility). *Suppose that **Assumption 3.3.2** is satisfied. Then the optimization problem (3.16) is recursively feasible for all $k \geq 0$ if it is feasible for $z(0) = x(0)$.*

Proof. Let $\mathbf{v}_k^* = \{v_{0|k}^*, \dots, v_{N-1|k}^*\}$ be the optimal solution sequence of (3.16) at time instant k , and the corresponding optimal state sequence is $\mathbf{z}_k^* = \{z_{0|k}^*, \dots, z_{N|k}^*\}$. Since $\mathbf{v}_k^*$ and $\mathbf{z}_k^*$ are feasible for the optimization problem (3.16), they satisfy the constraints (3.16f), (3.16g), and (3.16h). For the next time instant $k+1$, the candidate solutions $\mathbf{v}_{k+1}$ and $\mathbf{z}_{k+1}$ to the optimization problem (3.16) are chosen as

$$\mathbf{v}_{k+1} = \{v_{1|k}^*, \dots, v_{N-1|k}^*, v_{N-1|k+1}\}, \quad (3.17a)$$

$$\mathbf{z}_{k+1} = \{z_{1|k}^*, \dots, z_{N-1|k}^*, z_{N|k}^*, z_{N|k+1}\}, \quad (3.17b)$$

where $v_{N-1|k+1} = Kz_{N|k}^*$ and $z_{N|k+1} = A(\gamma_{N-1|k+1})z_{N|k}^* + B(\gamma_{N-1|k+1})Kz_{N|k}^*$ with $\gamma_{N-1|k+1} = h(z_{N|k}^*, Kz_{N|k}^*)$. It is clear that the first $N-1$ elements of $\mathbf{v}_{k+1}$ and the first N elements of $\mathbf{z}_{k+1}$ satisfy the constraints (3.16f) and (3.16g). Moreover, since $z_{N|k}^* \in \mathcal{Z}_f$, it follows

from **Assumption 3.3.2** that $v_{N-1|k+1} = Kz_{N|k}^* \in \mathcal{V}$ and $z_{N|k+1} = A(\gamma_{N-1|k+1})z_{N|k}^* + B(\gamma_{N-1|k+1})Kz_{N|k}^* \in \mathcal{Z}_f$, i.e., the last elements of $\mathbf{v}_{k+1}$ and $\mathbf{z}_{k+1}$ satisfy the constraints (3.16g) and (3.16h), respectively. According to the above analysis, it can be concluded that $\mathbf{v}_{k+1}$ and $\mathbf{z}_{k+1}$ are feasible for the optimization problem (3.16). $\square$

Despite being recursive feasible, this stochastic optimization problem above is computationally very expensive due to the endogenous scheduling variable that has nonlinear relationship with respect to system states and control inputs. To address this challenge and facilitate real-time implementations, we next present an efficient qLPV-SMPC algorithm via a shifted scheduling variable method.

3.4 Problem Reformulation for Efficient qLPV-SMPC

In this section, we reformulate the SMPC problem in (3.16) into a quadratic programming (QP) problem, which can be robustly and efficiently solved by many available solvers. The essential idea is to pre-calculate the endogenous scheduling variable sequence at each iteration by utilizing the qLPV system structure and the optimization history information, which will be detailed next.

As the nominal system (3.16b) does not include the system uncertainty $w_{i|k}$, we can directly expand (3.16b) along the prediction horizon as:

$$\begin{aligned} z_{1|k} &= A(\gamma_{0|k})z_{0|k} + B(\gamma_{0|k})v_{0|k}, \\ z_{2|k} &= A(\gamma_{1|k})A(\gamma_{0|k})z_{0|k} \\ &\quad + A(\gamma_{1|k})B(\gamma_{0|k})v_{0|k} + B(\gamma_{1|k})v_{1|k}, \\ &\dots \end{aligned} \tag{3.18}$$

$$z_{N|k} = \Psi_{N-1}^0 z_{0|k} + \sum_{i=0}^{N-1} \Psi_{N-1}^{i+1} B(\gamma_{i|k})v_{i|k},$$

where

$$\Psi_p^q = \begin{cases} \prod_{j=p}^q A(\gamma_{j|k}), & p \geq q, \\ I, & p < q. \end{cases}$$

By stacking the nominal states and the nominal control inputs into vector forms as $Z_k = [z_{1|k}^\top, z_{2|k}^\top, \dots, z_{N|k}^\top]^\top \in \mathbb{R}^{N \cdot n_x}$ and $V_k = [v_{0|k}^\top, v_{1|k}^\top, \dots, v_{N-1|k}^\top]^\top \in \mathbb{R}^{N \cdot n_u}$, (3.18) can be written as:

$$Z_k = M(\mathbf{\Gamma}_k) z_{0|k} + S(\mathbf{\Gamma}_k) V_k \quad (3.19)$$

where

$$M(\mathbf{\Gamma}_k) = \begin{bmatrix} A(\gamma_{0|k}) \\ A(\gamma_{1|k})A(\gamma_{0|k}) \\ A(\gamma_{2|k})A(\gamma_{1|k})A(\gamma_{0|k}) \\ \vdots \\ \Phi_{N-1}^0 \end{bmatrix}, \quad (3.20)$$

and

$$S(\mathbf{\Gamma}_k) = \begin{bmatrix} B(\gamma_{0|k}) & 0 & \cdots & 0 \\ A(\gamma_{1|k})B(\gamma_{0|k}) & B(\gamma_{1|k}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{N-1}^1 B(\gamma_{0|k}) & \Phi_{N-1}^2 B(\gamma_{1|k}) & \cdots & B(\gamma_{N-1|k}) \end{bmatrix}. \quad (3.21)$$

We similarly stack the expected error signals as $E_k = [\mathbb{E}[e_{1|k}^\top], \mathbb{E}[e_{2|k}^\top], \dots, \mathbb{E}[e_{N|k}^\top]]^\top$ and $E_k^0 = [\mathbb{E}[e_{0|k}^\top], \mathbb{E}[e_{1|k}^\top], \dots, \mathbb{E}[e_{N-1|k}^\top]]^\top$. From (3.16c) and $\mathbb{E}(w_{i|k}) = 0$, it then follows that

$$E_k = \begin{bmatrix} A_K(\gamma_{0|k}) & 0 & \cdots & 0 \\ 0 & A_K(\gamma_{1|k}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_K(\gamma_{N-1|k}) \end{bmatrix} E_k^0 = M_K(\mathbf{\Gamma}_k) e_{0|k}, \quad (3.22)$$

where $M_K(\mathbf{\Gamma}_k)$ has the same vector form as (3.20) with element $A(\cdot)$ replaced by $A_K(\cdot)$.

By using (3.19) and (3.22), the performance index (3.16a) can be reformulated into a compact form with V_k being the optimization variables. Meanwhile, the nominal state and nominal control input constraints (3.16f)~(3.16h) can be incorporated as a generalized matrix inequality constraint. These transformations cast the qLPV-SMPC problem (3.16)

to a compact matrix form as follows:

$$\min_{V_k} \quad J = V_k^\top L(\Gamma_k) V_k + 2q(\Gamma_k)^\top V_k \quad (3.23a)$$

$$s.t. \quad G(\Gamma_k) V_k \leq H(\Gamma_k), \quad (3.23b)$$

where $L(\Gamma_k) = S(\Gamma_k)^\top \bar{Q} S(\Gamma_k) + \bar{R}$, and $q = S(\Gamma_k)^\top (\bar{Q} M(\Gamma_k) z_{0|k} + \bar{Q} E_k) + \bar{R} \bar{K} E_k^0$.

The augmented diagonal weighting matrices $\bar{R}$, $\bar{K}$, and $\bar{Q}$ are given by $\bar{R} = \text{diag}_N(R)$, $\bar{K} = \text{diag}_N(K)$ and $\bar{Q} = \text{diag}(\text{diag}_{N-1}(Q), P)$. In (3.23b), $G(\Gamma_k)$ and $H(\Gamma_k)$ are derived from the constraints (3.16f)~(3.16h).

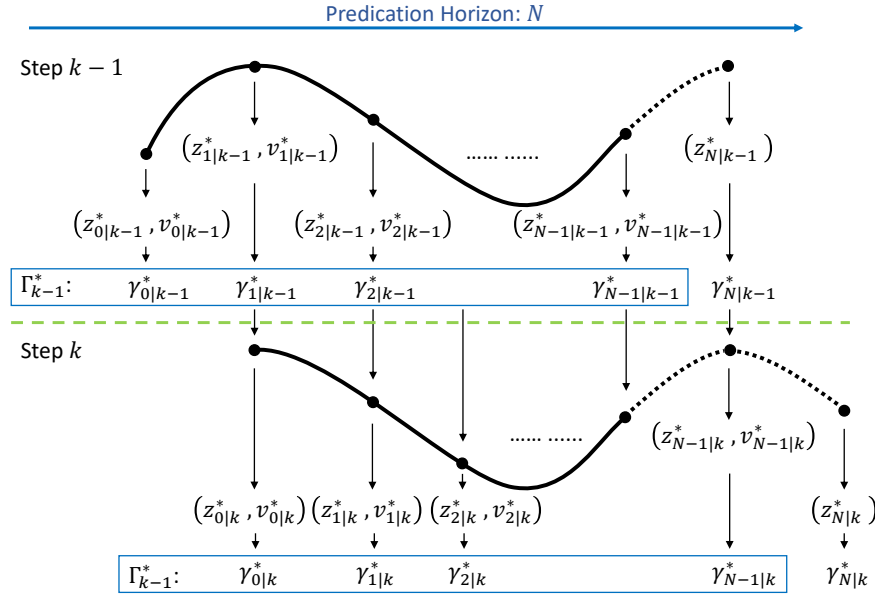


Figure 3.1 Scheduling variable and qLPV model updating strategy.

Note that if the scheduling variable sequence Γ_k is known as *a priori* or can be predicted, (3.23) then becomes a QP problem that can be efficiently solved, which is much more advantageous over directly solving a nonlinear optimization problem that has intractably computational complexity. As such, we develop the following shifted method for iterative Γ_k calculation and the qLPV structure updating:

- At the first sampling time, i.e., $k = 0$:

1. Initialize the scheduling variable $\gamma(0)$ based on the initial system state and a ‘guess’ control input according to $\gamma(0) = h(z(0), v(0))$.
 2. Compute M and S based on $\mathbf{\Gamma}_0 = \gamma(0) \otimes \mathbf{1}_N$.
 3. Solve the QP problem (3.23) and construct $\mathbf{\Gamma}_0^*$ via the obtained optimal nominal state sequence $\mathbf{z}_0^*$ and optimal nominal control input sequence $\mathbf{v}_0^*$.
- At each sampling time $k > 0$:
 1. Assign $\gamma_{i|k} = \gamma_{i+1|k-1}^*$ for $i = 0, \dots, N-2$. Then according to **Assumption 3.3.2** with $v_{N|k-1} = Kz_{N|k-1}^* \in \mathcal{V}$, construct $\mathbf{\Gamma}_k$ using $\gamma_{N-1|k} = h(z_{N|k-1}^*, v_{N|k-1})$. This strategy is illustrated in Fig. 3.1.
 2. Compute M , S , E_k and E_k^0 based on $\mathbf{\Gamma}_k$.
 3. Solve the QP problem (3.23) and construct $\mathbf{\Gamma}_k^*$ via the obtained optimal nominal state sequence $\mathbf{z}_k^*$ and optimal nominal control input sequence $\mathbf{v}_k^*$.

This results in solving a QP at each step and the steps are summarized in **Algorithm 3.4**.

Remark 3.4.1 (Calculation of $\gamma_{N|k}$). *In this strategy, the scheduling variables are computed as $\gamma_{i|k}^* = h(z_{i|k}^*, v_{i|k}^*)$ for $i = 0, \dots, N-1$. Based on **Assumption 3.3.2**, $\gamma_{N|k}^*$ is then approximated by $h(z_{N|k}^*, Kz_{N|k}^*)$. In the case that γ only depends on the system state z , i.e., $\gamma_{i|k}^* = h(z_{i|k}^*)$, no approximation is required to obtain $\gamma_{N|k}^*$.*

Remark 3.4.2 (Iterative Calculation of $\mathbf{\Gamma}_k$). *If computational capability allows, multi-iteration method [52] can be applied for iterative correction of the scheduling variable sequence, i.e., (3.23) can be solved multiple times as a series of QP sub-problems based on the updated $\mathbf{\Gamma}_k^m, m \geq 1$ at every iteration. The convergence of $(Z_k, V_k, \mathbf{\Gamma}_k) \rightarrow (Z_k^*, V_k^*, \mathbf{\Gamma}_k^*)$ of the QP problem can be achieved such that the recursive feasibility is retained.*

Algorithm 3.1 Quasi LPV Stochastic MPC.

Define the simulation length: N_{sim}

Define the weighting matrices: $\bar{Q}$ and $\bar{R}$

Stack the gain matrix $\bar{K}$;

Calculate the nominal state constraint: $\mathcal{Z} = \mathcal{X} \ominus \mathcal{R}$, the nominal control input constraint: $\mathcal{V} = \mathcal{U} \ominus \mathcal{R}_u$;

Initilize $z(0) = x(0)$ and $\mathbf{\Gamma}_0 = \gamma(0) \otimes \mathbf{1}_N$;

for $k = 0 \rightarrow N_{sim}$ **do**

/ Stochastic MPC */*

$z_{0|k} = z(k)$;

Generate M , S , E_k and E_k^0 based on $\mathbf{\Gamma}_k$;

Solve (3.23a) s.t. (3.23b) for V_k^* and Z_k^* ;

/ LPV system updating */*

Calculate $\mathbf{\Gamma}_k^* = h(Z_k^*, V_k^*)$ in element-wise;

Update $\mathbf{\Gamma}_{k+1} : \gamma_{i|k+1} = \gamma_{i+1|k}^*, i = 0, \dots, N-2, \gamma_{N-1|k+1} = h(z_{N|k}^*, K z_{N|k}^*)$ for next MPC iteration;

end

3.5 Case Study on Automotive Powertrain Control

This case study is a pilot demonstration of qLPV-SMPC on the engine system. Specifically, we present the implementation of the proposed qLPV-SMPC on an engine model that is obtained by eSTF with data collected from a naturally aspirated (NA) engine on a Ford F150 Truck. The schematic diagram of the NA internal engine is similar to 2.1 without a turbine (therefore, there is no wastegate actuator). The main actuators are throttle position, fuel injection port, intake cam position (ICAM), exhaust cam position (ECAM), and spark timing. To simplify the modeling procedure and inspired by [64], ICAM (u_1), ECAM (u_2), and throttle (u_3), which directly influence the NA engine breathing, are chosen as the engine inputs, while fuel consumption rate (x_1) and torque (x_2) are taken as the engine states.

3.5.1 Pseudo-Plant Construction From Vehicle Data

Based on the collecting data of the input-state signals, a set of local models (A_j, B_j) for the NA engine can be identified with eSTF based method [8]. The nonlinearity of the engine system is embedded in the scheduling variable $\gamma_j(\cdot)$, which belongs to a unit simplex Λ . More precisely, $\gamma_j(\cdot)$ is a softmax-like function described by

$$\gamma_j = \frac{\exp\{\mathcal{D}((x, u), \mathcal{C}_j)\}}{\sum_{l=1}^{n_\gamma} \exp\{\mathcal{D}((x, u), \mathcal{C}_l)\}} \in \Lambda, j = 1, \dots, n_\gamma, \quad (3.24)$$

where $x \triangleq [x_1 \ x_2]^\top$, $u \triangleq [u_1 \ u_2 \ u_3]^\top$, $\mathcal{D}(\cdot)$ is a dissimilarity metric by taking into account the clustering information $(\mathcal{C})$ with respect to input-output data. For the dynamic construction of $\mathcal{D}$, readers can refer to [7] for more details. In addition, stochastic disturbance w in the following form is added to the qLPV model to characterize the lumped model uncertainty and additive disturbance:

$$w \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \text{Var}(w_f) & 0 \\ 0 & \text{Var}(w_t) \end{bmatrix}\right), \quad (3.25)$$

where $\text{Var}(w_f)$ and $\text{Var}(w_t)$ are the disturbance variance with respect to the fuel consumption rate and torque signals.

3.5.2 Performance Index and System Constraints

Based on the system structure and fuel economy purpose, we consider the following optimization objectives:

- Objective 1: increase fuel economy (reduce fuel consumption rate (x_1)).
- Objective 2: keep engine torque (x_2) tracking a desired profile.

The corresponding stage cost and terminal cost functions are defined as

$$\ell(x_{i|k}, u_{i|k}) = \|x_{i|k} - r(k+i)\|_Q^2 + \|u_{i|k}\|_R^2, \quad (3.26a)$$

$$V_f(x_{N|k}) = \|x_{N|k} - r(k+N)\|_P^2, \quad (3.26b)$$

where $r(k+i) \triangleq \begin{bmatrix} r^{x_1}(k+i) & r^{x_2}(k+i) \end{bmatrix}^\top \in \mathbb{R}^2$ is the given state reference. For fuel minimization objective, we set $r^{x_1} = 0$; while for torque tracking objective, r^{x_2} is a given ideal torque profile.

Based on the aforementioned definition, the following performance index is adopted by MPC at each time instant k :

$$J = \mathbb{E} \left[\sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}) + V_f(x_{N|k}) \right] + \rho \varepsilon_k^2, \quad (3.27)$$

where $\varepsilon_k \in \mathbb{R}$ is a slack variable for feasible solution. To ensure safe and efficient operations, the system constraints for this engine plant are given by

$$\Pr(x_{\min} \leq x(k)) > p_x, \quad (3.28a)$$

$$\Pr(u_{\min} \leq u(k) \leq u_{\max}) > p_u, \quad (3.28b)$$

$$-\Delta u_{\max} - \varepsilon_k I \leq \Delta u(k) \leq \Delta u_{\max} + \varepsilon_k I, \quad (3.28c)$$

where $\Delta u(k) = u(k) - u(k-1)$. The constraints for all states and control inputs are chosen based on real vehicle testing experience. Here p_x is tunable for different requirements, while due to physical actuator limitation, a quite hard chance constraints on the control inputs ($p_u = 0.99$) is preferred. For the same purpose, we also include an incremental rate constraint (3.28c).

By using the PRS techniques discussed in Section 3.3.1, the chance constraints (3.28a) and (3.28b) on the engine model can be transferred to deterministic constraints on the nominal system state z and input v . The incremental rate constraint on $\Delta u(k)$ is applied on $\Delta v(k)$ directly. After handling the state and input constraints, a SMPC problem similar to (3.16) can be formulated. The slack variable ε_k is augmented with the nominal control input vector, i.e.,

$$\tilde{V}_k = \begin{bmatrix} V_k \\ \varepsilon_k \end{bmatrix}. \quad (3.29)$$

Following Section 3.4 and the technique used in [65, p.6], the SMPC problem can be rewritten as a compact matrix form:

$$\begin{aligned} \min_{\tilde{V}_k} \quad & J_{QP} = \tilde{V}_k^\top \tilde{L} \tilde{V}_k + 2\tilde{q}^\top \tilde{V}_k, \\ \text{s.t.} \quad & \tilde{G} \tilde{V}_k \leq \tilde{H}, \end{aligned} \tag{3.30}$$

where matrices $\tilde{L}, \tilde{q}, \tilde{G}$ and $\tilde{H}$ are modified versions of L, q, G and H in (3.23).

3.5.3 Simulation Results of qLPV-SMPC on NA Engine Model

This subsection discusses the qLPV-SMPC performance on the identified NA engine model. The penalty matrix Q for the performance of each objective can be tuned and initialized by pareto front [66] for a purpose that the minimization of fuel consumption should be guaranteed under a good torque tracking condition. Meanwhile R is usually chosen small to provide enough freedom for MPC optimization. In this case study, the matrices Q, R, P and the penalty parameter for slack variable are chosen as:

$$Q = \begin{bmatrix} 0.124 & 0 \\ 0 & 1.64 \end{bmatrix}, \quad R = 1e^{-2}I_3, \quad P = Q, \quad \rho = 1.$$

Based on this setting, qLPV-SMPC is solved with $p_x = 0.85$ and $p_u = 0.99$. The total fuel consumed for this period is accumulated up to 54.332, which is a scaled value¹, but reasonable if compared to the scaled real fuel consumption calculated from vehicle data with the same torque profile. On the other hand, the evolution of the torque trajectory and the reference one is shown in Fig. 3.2. The normalized root-mean-square-error (NRMSE) between the torque and the reference is 0.097034, which indicates a good satisfaction of torque tracking objective.

The PRS for $p_x = 0.85$ and the distribution of system error $e(k) = x(k) - z(k)$ during the whole MPC procedure are presented in Fig. 3.3. The dot markers within the PRS represent the constraint satisfaction of the actual system state, while the cross markers outside the

¹The real value not presented for confidential reason.

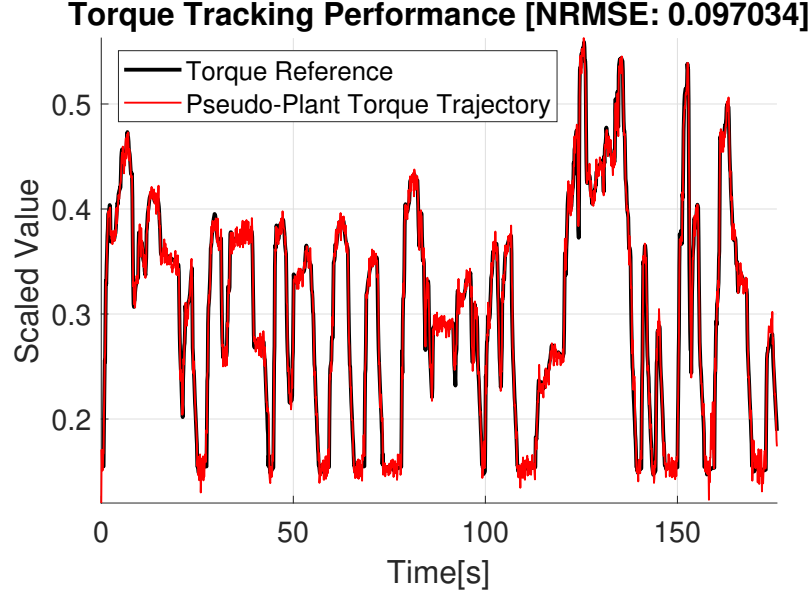


Figure 3.2 Torque tracking performance.

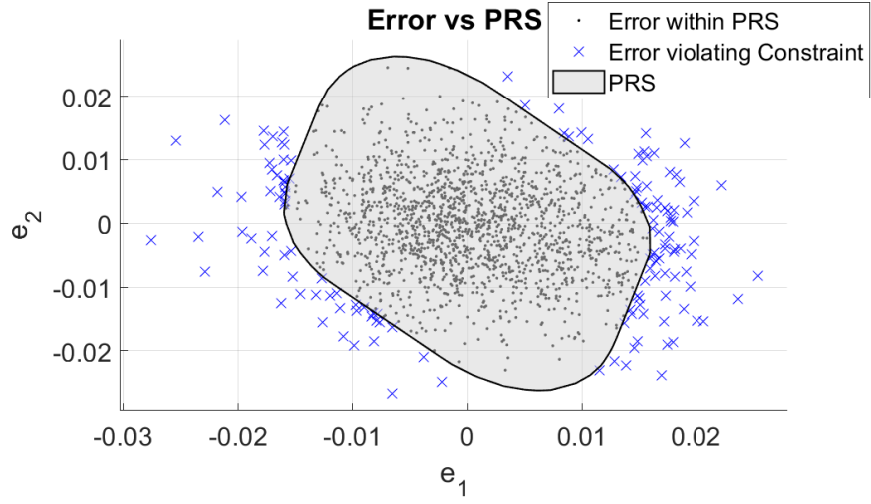


Figure 3.3 State error $e(k)$ distribution during the whole MPC optimization process and the PRS $\mathcal{R}^x$ with $p_x = 0.85$.

PRS indicate constraint violation. The percentage of the system error within the PRS is 90.04%, implying the chance constraints are satisfied with the aid of PRS techniques.

The dynamic evolution of the real and nominal system states during a segment of qLPV-SMPC process (time interval: $148s \sim 154s$) is shown in Fig. 3.4. The cross-section of PRS tube is centered by the nominal state trajectory, while the actual state is bounded by the

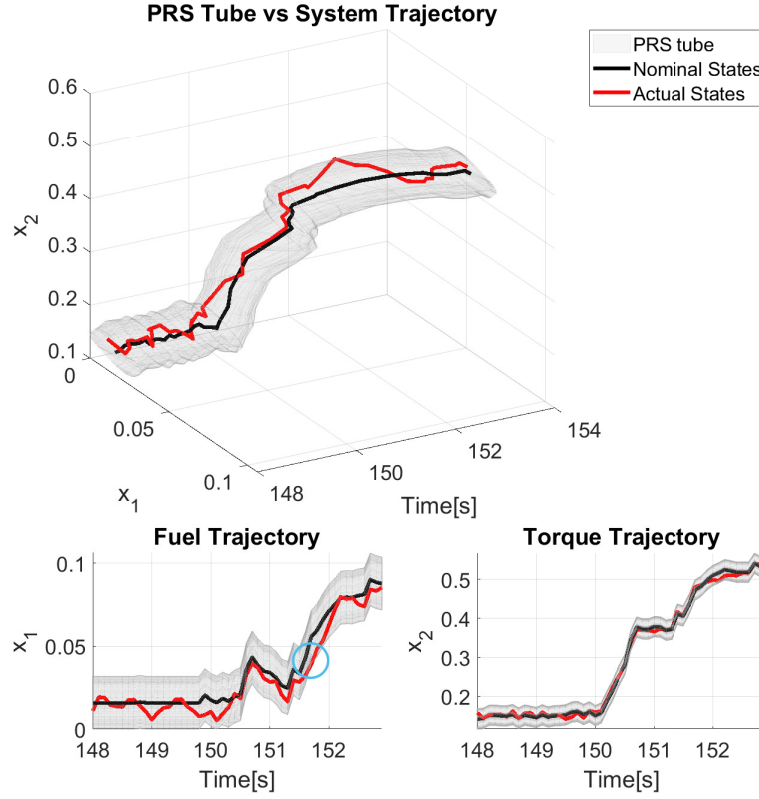


Figure 3.4 A segment of MPC optimization process. Top: a segment of 3D PRS tube and corresponding system trajectories; Left-Bottom: Projection onto $x_1 - t$ plane; Right-Bottom: Projection onto $x_2 - t$ plane. The blue circle shows a constraint violation point.

PRS tube with pre-defined probability level. A constraint violation point is marked by a blue circle to indicate the PRS nature. Different from the robust reachable set, the PRS is used as a probabilistic tube to confine the system dynamics and will lead to less conservatism on the system operation.

3.5.4 Comparison between qLPV-RMPC and qLPV-SMPC

In this subsection, RMPC is applied to the same qLPV system to facilitate the comparison. For the RMPC problem, hard constraints on the system state and control input are considered, and the minimal robust positively invariant set (mRPIS) defined in [60], [67] is used to tighten the constraints under bounded disturbance. mRPIS is calculated with the method

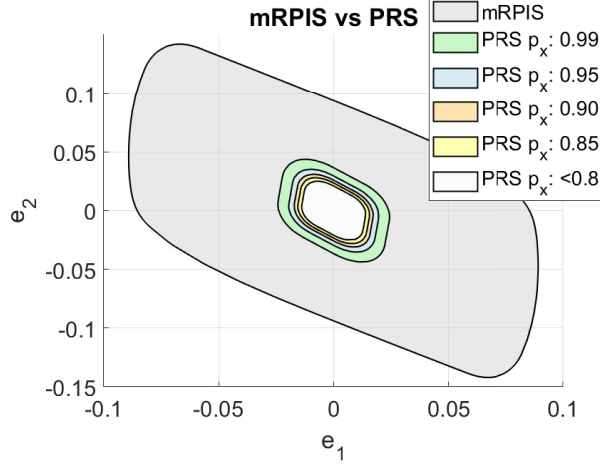


Figure 3.5 Comparison of tube size: mRPIS vs. PRS.

introduced in [68].

In Fig. 3.5, mRPIS and PRSs under different probabilities are compared under the framework of RMPC and SMPC, respectively. It is clear that large mRPIS will lead to a restricted feasible region for the nominal state and input, while small PRSs can alleviate the restriction. The influence of such a region limitation for control performance is shown in Fig. 3.6. It can be seen that the qLPV-RMPC cannot effectively track the torque reference compared to qLPV-SMPC with different PRSs. As presented in Fig. 3.7, the control inputs of qLPV-RMPC (red line) are confined in a smaller region compared to the ones of qLPV-SMPC.

Table 3.1 Comparison of total fuel consumption of qLPV-RMPC and qLPV-SMPC.

Total Fuel Consumed (scaled values)				
qLPV-RMPC	qLPV-SMPC			
	$p_x : 0.99$	$p_x : 0.95$	$p_x : 0.90$	$p_x : 0.85$
167.7767	61.1058	57.4073	54.6989	52.0954

On the other hand, the comparison results about the fuel consumption are summarized in Table 3.1. Compared to SMPC method, RMPC scheme leads to much higher fuel consumption. In addition, for the SMPC method, lower probability p_x means more constraint violation but will lead to better fuel economy achievement to some extent.

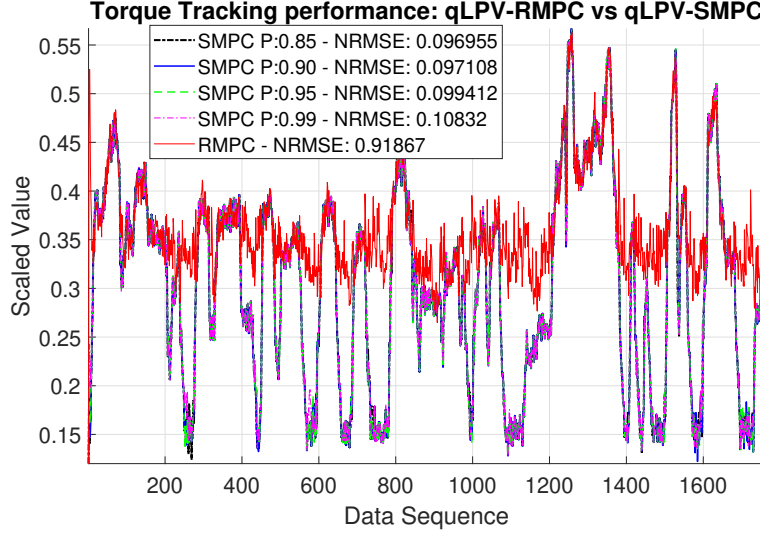


Figure 3.6 Comparison between qLPV-RMPC and qLPV-SMPC: torque tracking.

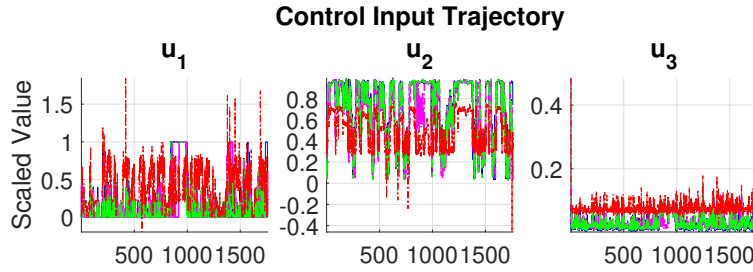


Figure 3.7 Comparison between qLPV-RMPC and qLPV-SMPC: control trajectory.

3.6 Conclusion

To summarize this chapter, based on our previous work for nonlinear system identification, we proposed an efficient qLPV-SMPC method for nonlinear systems subject to stochastic disturbance and chance constraints. The probabilistic description of disturbance was systematically incorporated into the MPC problem formulation after tightening the constraints with PRS. With the shifted scheduling variable method, the nonlinear optimal control problem was transformed to a series of computation-tractable QP problems. A case study was demonstrated by applying the developed qLPV-SMPC scheme to the automotive engine model. The results showed that the qLPV-SMPC scheme can accomplish the multi-objective target while satisfying the specified chance constraints. Compared with qLPV-RMPC, the

less conservative feature further shown the effectiveness of the proposed method in the engine model case. In the future, we will implement the developed scheme in a real engine system and conduct a thorough validation.

CHAPTER 4

EFFICIENT AND PRIVACY-PRESERVING DATA-ENABLED PREDICTIVE CONTROL FOR AUTOMOTIVE SYSTEMS

4.1 Introduction of Data-Enabled Predictive Control

Data-Enabled Predictive Control (DeePC) has recently emerged as a promising model-free optimal control paradigm that directly uses input-output data for controlling systems with unknown dynamics [3]. In contrast to traditional model-based control schemes that rely on an accurate parametric model [65], DeePC leverages behavioral system theory [6] and Willems’ fundamental lemma [69] to implicitly describe the system trajectories using previously collected input-output data [3]. Its receding horizon implementation is shown to be equivalent to a model predictive control (MPC) formulation for linear time-invariant (LTI) systems [70], while its application to nonlinear and stochastic systems also shows promising performance via different regularization techniques [71], [72].

Despite its promise, application of DeePC to real-time engineering systems still poses significant challenges. First, in order to satisfy the persistent excitation (PE) condition, DeePC needs to solve an optimization problem with a large dimension of optimization variables (typically larger than those in its MPC counterpart), making it computationally expensive for real-time implementation. It is thus crucial to develop efficient approaches to reduce its computation burden and enable its practical use and wider adoption in real-world engineering systems. Second, another important factor hindering DeePC’s application in real-world cyber-physical systems (CPS) is privacy concern due to DeePC’s heavy reliance on the intensive collection and sharing of data, particularly in a networked, multi-agent setting with close interaction or cooperation. In fact, privacy concerns on data collection and sharing in CPS have severely hindered networked robot deployment in European urban areas [73] and government drone use in North America [74]. It is worth noting that conventional privacy

mechanisms either trade accuracy for privacy (e.g., differential privacy [75]) or incur heavy computation/communication overhead (e.g., crypton [76]), and hence are inappropriate for CPS subject to stringent accuracy and real-time constraints (for example, reported homomorphic encryption based privacy solutions for self-driving cars incur computation latencies on the order of *seconds* [77]–[85] whereas real-time formation control can only afford latency on the order of *milliseconds* [86], [87]).

To address the aforementioned challenges, the principal research goal of this chapter is to develop a comprehensive, unifying framework and corresponding novel methodologies for efficient and privacy-preserving data-enabled predictive control. To pursue the above goal, the specific tasks of this chapter include:

- Develop robust dimension reduction method to significantly improve the computational efficiency while maintaining the PE conditions, enabling DeePC’s practical application with limited computation and real-time constraints.
- Formalize and synthesize novel algorithm-intrinsic privacy-preserving schemes, to protect privacy in DeePC without sacrificing accuracy or incurring heavy computation or communication overhead.

4.2 Preliminaries

DeePC is developed under LTI systems as a model-free optimal control strategy that does not rely on an explicit parametric model. Instead, it leverages Willems’ fundamental lemma [69] and generates control inputs (and resultant system outputs) compatible with the pre-collected system data. Specifically, let $u^d = [u_1^d, u_2^d, \dots, u_m^d]^T \in \mathbb{R}^m$ be the control inputs and let $y^d = [y_1^d, y_2^d, \dots, y_r^d]^T \in \mathbb{R}^r$ be the system outputs. As shown in Fig. 4.1, an input/output data sequence of length T is collected, and the consecutive trajectories of length $T_{\text{ini}} + N$ are extracted to form the following Hankel matrices:

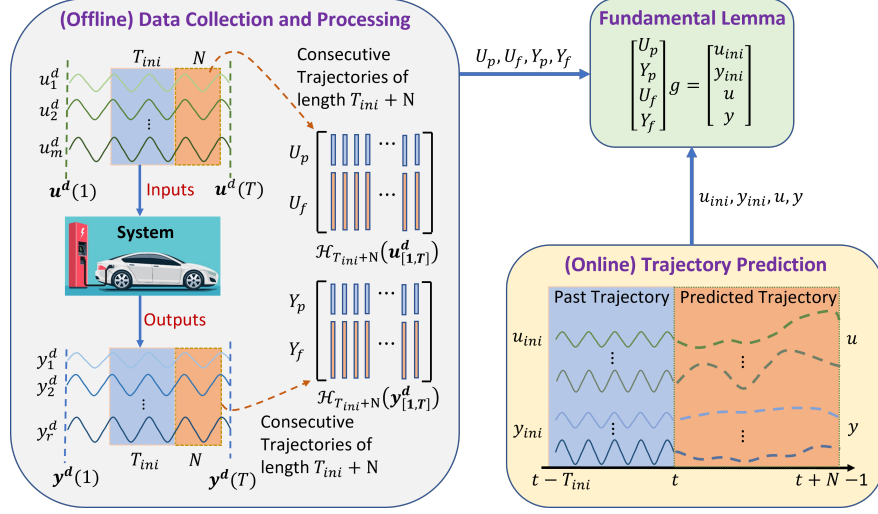


Figure 4.1 Schematic diagram of the data-enabled predictive control.

$$\begin{aligned}
 \mathcal{H}_L(u_{[1,T]}^d) &= \begin{bmatrix} u^d(1) & u^d(2) & \cdots & u^d(T-L+1) \\ u^d(2) & u^d(3) & \cdots & u^d(T-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ u^d(L) & u^d(L+1) & \cdots & u^d(T) \end{bmatrix} = \begin{bmatrix} U_p \\ U_f \end{bmatrix}, \\
 \mathcal{H}_L(y_{[1,T]}^d) &= \begin{bmatrix} y^d(1) & y^d(2) & \cdots & y^d(T-L+1) \\ y^d(2) & y^d(3) & \cdots & y^d(T-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ y^d(L) & y^d(L+1) & \cdots & y^d(T) \end{bmatrix} = \begin{bmatrix} Y_p \\ Y_f \end{bmatrix}.
 \end{aligned} \tag{4.1}$$

Where $L = T_{ini} + N$. U_p denotes the first T_{ini} block rows of $\mathcal{H}_L(u_{[1,T]}^d)$ and represents the “past” segment of the input trajectory whereas U_f denotes the last N block rows of $\mathcal{H}_L(u_{[1,T]}^d)$ and represents the “future” segment of the input trajectory. The matrices Y_p and Y_f are similarly defined. Essentially, the columns of $\mathcal{H}_L(u_{[1,T]}^d)$ and $\mathcal{H}_L(y_{[1,T]}^d)$ can be viewed as libraries of input/output trajectories of length L , which are further partitioned into segments with length T_{ini} and length N to be compatible with the online optimization formulation which will be discussed later. Beforehand, a concept needs to be presented:

Definition 4.2.1 (Persistently Exciting). *A signal sequence $w_{[1,T]}$ is persistently exciting of order L ($L \leq T$) if the Hankel matrix $\mathcal{H}_L(w_{[1,T]})$ is of full row rank.*

Remark 4.2.2 (Minimal Length of T). *In order for $w_{[1,T]}$ to be persistently exciting of order L , it must have $T \geq (m+1)L-1$, that is, the input signal sequence $w_{[1,T]}$ should be sufficiently rich and long as to excite the system yielding an output sequence that is representative for the system's behavior.*

During the online implementation, at each time step t , input/output trajectory of the past T_{ini} steps is buffered and used to form $u_{\text{ini}} = u_{[t-T_{\text{ini}}, t-1]}$ and $y_{\text{ini}} = y_{[t-T_{\text{ini}}, t-1]}$. Let $u = u_{[t, t+N-1]}$ and $y = y_{[t, t+N-1]}$ be the control inputs and system outputs over the N -step prediction horizon, respectively. The Willems' fundamental lemma states that if the pre-collected input sequence $u_{[1,T]}^d$ is persistently exciting of order $T_{\text{ini}} + N + n$ with n being the dimension of the system states (n can be chosen as an upper bound of state dimension [3]), then the patched trajectory $(u_{\text{ini}}, y_{\text{ini}}, u, y)$ is generated from the system when it is spanned by (U_p, Y_p, U_f, Y_f) , that is, there exists a vector $g \in \mathbb{R}^{T-T_{\text{ini}}-N+1}$ such that:

$$\begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}, \quad (4.2)$$

which is a non-parametric system representation that ensures any consecutive trajectory of length $T_{\text{ini}} + N$ is compatible with those in the trajectory library in (4.1), and the initial trajectory $[u_{\text{ini}}^\top, y_{\text{ini}}^\top]^\top$ can be thought of as setting an initial condition for the unknown future trajectory $[u^\top, y^\top]^\top$. The DeePC can then be cast as the following constrained optimization

problem:

$$\begin{aligned}
\min_{g,u,y} \quad & J(u, y) = \sum_{k=t}^{t+N-1} \left(\|y(k)\|_Q^2 + \|u(k)\|_R^2 \right) \\
\text{s.t.} \quad & (4.2), \\
& y(k) \in \mathcal{Y}(k), \quad k = t, \dots, t+N-1, \\
& u(k) \in \mathcal{U}(k), \quad k = t, \dots, t+N-1,
\end{aligned} \tag{4.3}$$

where $\mathcal{U}(k)$ and $\mathcal{Y}(k)$ represent the input and output constraints at time step k , respectively. As DeePC bypasses the model identification step and directly optimizes the control from data, it has found great successes across various domains [3], [88]–[90].

DeePC is developed under deterministic LTI system, beyond which a regularized DeePC scheme provides insightful extension to nonlinear system control [3], [71], [91], [92].

As the input/output data are collected from nonlinear system, the subspace spanned by the constructed Hankel matrix may no longer be consistent with the subspace of trajectories of the underlying system, even if the Hankel matrix is easy to get full rank under such data sets collected from nonlinear system. A direct consequence is that poor prediction performance will be achieved by the ill-posed condition of Hankel matrix constraint. In this case, a penalty term is added for the difference, quantified by a slack variable σ_y , between estimated initial condition $Y_p g$ and the real-time buffered initial condition y_{ini} , which provides a least-square estimation of the true initial condition. Then with an additional regularization

term, the corresponding regularized DeePC optimization problem is presented as:

$$\min_{g,u,y} J(u, y) + \lambda_y \|\sigma_y\|_2^2 + \lambda_g \|g\|_2^2 \quad (4.4a)$$

$$\text{s.t.} \quad \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix}, \quad (4.4b)$$

$$y(k) \in \mathcal{Y}(k), \quad k = t, \dots, t + N - 1, \quad (4.4c)$$

$$u(k) \in \mathcal{U}(k), \quad k = t, \dots, t + N - 1, \quad (4.4d)$$

where $\lambda_y, \lambda_g \in \mathbb{R}_+$ are regularization parameters. This problem formulation (4.4) has been proved to coincide with a distributionally robust problem via the technique of Wasserstein metric [71], [91].

4.3 Efficient DeePC for Lithium-Ion Battery Fast Charging

4.3.1 Introduction of Lithium-ion Battery Fast Charging Problem

Due to clean and energy-efficient properties [93], lithium-ion (Li-ion) battery has taken the forefront as power source for electronic vehicles (EVs). However, compared to gasoline vehicles with quick refueling, long charging time is regarded as the major hurdle to the widespread commercialization of Li-ion battery-powered EVs. Conventionally, the charging time of Li-ion battery can be reduced by simply raising the charging rate, which, however, may intensify the battery's degradation and shorten its lifespan. Therefore, the development of fast charging approaches for Li-ion batteries should not only address the charging time issue but also maintain the battery in a safe and nondestructive charging mode.

Currently, constant-current and constant-voltage (CC/CV) [94] is the most commonly used charging protocol for Li-ion batteries in industry. It consists of two charging phases: a CC charging phase in which battery voltage raises to a predetermined value; and a CV charging phase in which voltage remains constant until current falls below a predetermined

threshold. In addition, some variants, such as multistage constant current protocols [95], [96] and pulse charging protocols [97], [98], are designed to achieve faster charging and/or less battery degradation. However, it is challenging and empirical to determine and design the charging profiles for these protocols. Moreover, they are unable to explicitly handle system constraints and not optimal with regard to charging time, safety and degradation.

To overcome the aforementioned difficulties, model-based optimization methods have been developed. The commonly seen battery models can be roughly classified into electric equivalent circuit models (EECMs) and electrochemical models (EMs). EECMs are composed by different electrical components, e.g., voltage source, resistances and capacities. With parameterization and identification processes, EECMs are able to estimate several state characteristics of batteries, such as state-of-charge (SOC) [99], [100], electrode voltage [101], impedance [100] and heat generation [102]. Based on the obtained state characteristics, the optimization techniques, such as dynamic programming [103], generic algorithm [102], [104], fuzzy control [105], and min-max strategy [106], can be designed to generate optimal charging protocols for fast charging, while mitigating battery degradation. However, EECMs ignore the description about certain internal battery dynamics, especially for side reactions caused by chemical process, and thus their applications are limited by low modeling accuracy, particularly at a wide range of operating conditions.

Conversely, the EMs [107] are designed based on first-principle modeling method and can capture the internal mechanisms, e.g., electrochemical reactions, ion diffusion and de-/intercalation processes. Therefore, it has high fidelity for battery behavior description. However, EMs are complex to establish and calibrate [108], so that the controller synthesis is very challenging. Attempts have been made to reduce EM complexity by the simplification of battery structure [109] and internal dynamics [110]. With the simplified models, many closed-loop optimal controls are formulated for battery fast charging. For instance, a fast charging method was developed in [111] based on simplified isothermal EM, side reaction rate model, and extended Kalman filter-based CC/CV protocol. By incorporating thermal dynamics into

similar models, a reference governor based nonlinear model predictive control (MPC) [112] was proposed to reduce charging time and mitigate the impact of temperature variation on battery degradation. Although model-based approaches provide optimal solutions for fast charging in theory, robustness and optimality can rarely be accomplished in practice because the Li-ion battery dynamics are ever-changing due to distinct operation modes and uncertainties.

These observations, along with improvements in computing and sensing technology, have motivated a trend toward model-free and data-driven control techniques. In particular, Data-Enabled Predictive Control (DeePC) recently emerged as a promising model-free optimal control paradigm that directly uses input-output data to achieve safe and optimal control of unknown systems [3]. In contrast to traditional model-based control schemes that rely on an accurate parametric model [65], DeePC leverages behavioral system theory [6] and Willems’ fundamental lemma [69] to implicitly describe the system trajectories using collected input-output data [3]. It has been revealed that DeePC is equivalent to an MPC formulation for linear time-invariant (LTI) systems [70], and its application to nonlinear and stochastic systems shows promising performance with the aid of different regularization techniques [71], [72]. The DeePC is especially appealing to systems where the model is difficult or expensive to obtain, and it has been successfully implemented in different applications, including unmanned aerial vehicles [3], power grids [89], and connected and automated vehicles [90]. However, the huge dimension of optimization variables in DeePC (usually larger than those in its MPC counterpart) makes it computationally expensive for real-time implementation. Therefore, it is crucial to develop effective methods to reduce computational cost, enabling its application and widespread adoption in actual engineering systems.

Considering the aforementioned concerns, this chapter proposes an efficient DeePC approach for fast charging of Li-ion batteries with constraint handling. Specifically, without relying on a complex battery model, an optimal charging protocol is designed by utilizing measurable battery input and output data. Both input current and battery internal state

constraints are explicitly considered in the DeePC formulation, ensuring a safe charging operation. Meanwhile, a singular value decomposition (SVD) [113] based method is proposed to reduce the dimension of optimization variables and thus alleviate the computational burden. Our main contributions include: 1) To the best of our knowledge, this is the first time the DeePC framework is used to address the issue of safe and fast charging of Li-ion batteries. The cumbersome modeling and validation processes of EECMs and EMs are circumvented by offline data measurement and processing. 2) We propose a novel SVD-based dimension reduction method for the ease of DeePC online implementation, while still maintaining the optimal performance of original DeePC. 3) Finally, the efficacy of the proposed scheme on fast charging of Li-ion batteries with safety related constraints is evaluated by simulations, showing that the developed method not only achieves safe and fast charging, but is also lightweight in computation.

The rest of this chapter is organized as follows. Section 4.3.2 introduces the DeePC algorithm for optimal charging of Li-ion battery systems. Section 4.3.3 presents the SVD-based dimension reduction method for the optimization variables of DeePC. Section 4.3.4 shows the simulation results, and conclusions are made in Section 4.3.5.

Nomenclature: We denote $\mathbb{Z}_+$ and $\mathbb{R}$ as the set of positive integers and real numbers, respectively. $\mathbf{0}$ used in $[\cdot]$ is denoted as a zero vector with compatible size. The weighted 2-norm of a vector x is denoted by $\|x\|_P \triangleq \sqrt{x^\top P x}$, where $P \succ (\succcurlyeq) \mathbf{0}$ is a positive (semi-)definite matrix. Given a signal $\omega \in \mathbb{R}^n$ and two integers $i, j \in \mathbb{Z}_+$ with $i \leq j$, we denote by $\omega_{[i,j]}$ the restriction of ω to the interval $[i, j]$, namely, $\omega_{[i,j]} := \left[\omega^\top(i), \omega^\top(i+1), \dots, \omega^\top(j) \right]^\top$. To simplify notation, we will also use $\omega_{[i,j]}$ to denote the sequence $\{\omega(i), \dots, \omega(j)\}$. Given a matrix $A \in \mathbb{R}^{n \times m}$, we denote by $A_{[i,j]}$, $1 \leq i < j \leq m$ the restriction of A to the interval from i th column to j th column. We refer to $x(t)$ the system measurement at time instant t , while quantity $x_{i|t}$ is a prediction value i time-steps ahead of current time instant t .

4.3.2 DeePC Formulation for Fast Charging of Lithium-ion Battery

In this section, we first define the working conditions of Li-ion battery to avoid unnecessary chemical complexity as those in EECMs and EMs for the ease of DeePC implementation. Then, a non-parametric representation for Li-ion battery charging dynamics will be formed based on system measurements. Finally, we discuss the objective and system constraints to formulate the overall DeePC problem for Li-ion battery fast charging.

4.3.2.1 Working Conditions for Lithium-ion Battery Charging

The working mechanism of Li-ion battery is vary complex with combined electrical, chemical and thermal dynamics, and they are interacted to influence the battery performance. For instance, ion diffusion rate varies under different temperatures, while ion with different kinetics consumed by chemical reaction leads to temperature fluctuation. So improper charging condition may lead a Li-ion battery to unstable and even unsafe region. So for safely charging purpose, in this research, extreme charging scenarios are not considered, that is, 1) the charging rate will be maintained under $2C$. Because rapid heat generation [111] will be induced by high charging rate, causing battery degradation, and terminal voltage will also increase fast [114] to cut-off value, leading to an early termination of battery charging. 2) The limits of lower and upper cut-off voltages are provided to stop the charging process to avoid over-powered charging beyond safe charging zone, even if it may be a little bit conservative [101]. 3) The operating temperature domain is also constrained between $25^{\circ}C$ and $31^{\circ}C$ to neglect the effect of Li-ion plating [111], [112], [115]. All these limits are geared towards demonstrating the benefits of DeePC for Li-ion battery fast and safe charging purposes, without considering negligible battery internal side reactions. In the long run, however, the side reactions may cause irresistible degradation to the Li-ion battery, and lots of works are focusing on the avoidance and/or alleviation of such side reactions, so the safe and fast charging of Li-ion battery, with minimized side reaction effect, is still an open research field till now and in the near future.

4.3.2.2 Non-Parametric Representation of Lithium-Ion Battery Charging Dynamics

Conventional Li-ion battery fast charging methods rely on explicit system models [116], i.e. EECMs and EMs, to facilitate controller design. One typical example refers to MPC [112], [114], [117], [118] whose performance depends closely on battery model accuracy. Although there exist many system identification techniques, obtaining a high fidelity model to capture battery dynamics is still difficult and even impossible without simplifications or assumptions. DeePC [3], therefore, is a promising model-free paradigm that can be applied on battery system for fast charging purpose. In particular, this data-driven method can directly describe the explicit dynamics of batteries in a non-parametric fashion based on Willems' fundamental lemma [69] and variable regularization.

Next, the non-parametric representation of battery charging dynamics will be discussed. Data collection is the first step. Specifically, a trajectory of length $T \in \mathbb{Z}_+$ is collected during battery charging phase, and the trajectory includes the following two parts:

$$u_{[1,T]}^d = \begin{bmatrix} u^d(1) \\ \vdots \\ u^d(T) \end{bmatrix} \in \mathbb{R}^{mT}, \quad y_{[1,T]}^d = \begin{bmatrix} y^d(1) \\ \vdots \\ y^d(T) \end{bmatrix} \in \mathbb{R}^{rT};$$

with $u_{[1,T]}^d$ being the input charging current (I) sequence and $y_{[1,T]}^d$ the output sequence whose element $y(i) = [y_V(i), y_T(i), y_{soc}(i)]^\top, i \in [1, T]$, shown in Fig. ??, representing battery voltage (V), battery temperature (T) and SOC. Note that SOC cannot directly be measured in reality, however, it is often tracked by means of current integration [108], assuming precise current measurements and correct initial conditions. So SOC can also be taken as a measurement through calculation in real plant. These pre-collected data are stored and re-arranged as the Hankel matrix [3] for non-parametric representation. Then, the Hankel matrix is further partitioned into two blocks based on (4.1), corresponding to the “past” data

of length T_{ini} and “future” data of length N data, respectively:

$$\begin{bmatrix} U_{\text{p}} \\ U_{\text{f}} \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (u_{[1,T]}^{\text{d}}), \quad \begin{bmatrix} Y_{\text{p}} \\ Y_{\text{f}} \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (y_{[1,T]}^{\text{d}}). \quad (4.5)$$

Remark 4.3.1 (Construction of Hankel matrix). *If a system is undesirable to generate a long trajectory, e.g. the system has unstable dynamics or expensive to do so. In this case, multiple short system trajectories can be used instead of a long one to construct the Hankel matrix, as long as it satisfies a collective persistency of excitation condition [119].*

Finally, with the buffered $u_{\text{ini}} = u_{[t-T_{\text{ini}}, t-1]}$ and $y_{\text{ini}} = y_{[t-T_{\text{ini}}, t-1]}$ of length T_{ini} and also the defined $u = u_{[t, t+N-1]}$ and $y = y_{[t, t+N-1]}$ of length N , the system behavior can be represented as the form (4.4b) at the given time instant t , based on Willems’ fundamental lemma and the regularization technique.

4.3.2.3 DeePC Formulation for Battery Fast Charging

The charging strategy proposed in this chapter aims to charge a Li-ion battery as fast as possible, at the same time, in a safe manner. Specifically, the objective is to design an optimal charging current profile to complete the charging task with minimum time consumed, while satisfying the aforementioned working condition constraints. To facilitate the practical application, instead of using the total time as a cost during the whole charging process, the objective function is designed by driving SOC to a target value. In addition, to alleviate the current chattering phenomenon and maintain a smooth charging process, the input variation in successive optimization steps are penalized. Then, the cost function is designed as,

$$\min_{g, u, y} J(u, y) = \sum_{k=t}^{t+N-1} \left(\|y_{\text{soc}}(k) - r_{\text{soc}}\|_Q^2 + \|\Delta u(k)\|_R^2 \right), \quad (4.6)$$

where Q and R are penalty matrices. By adding the regularization terms for DeePC extension to nonlinear system, the cost function is expanded to be (4.4a). To maintain the battery

charging problem within a safe region, the following constraints are required to be satisfied:

$$\begin{aligned}
I_{min} &\leq u(k) \leq I_{max}, \\
\Delta I_{min} &\leq \Delta u(k) \leq \Delta I_{max}, \\
V_{min} &\leq y_V(k) \leq V_{max}, \\
T_{min} &\leq y_T(k) \leq T_{max}, \\
SOC_{init} &\leq y_{soc}(k) \leq SOC_{final},
\end{aligned} \tag{4.7}$$

where $I_{min/max}$ and $\Delta I_{min/max}$ denote the lower and upper bounds of input charging current and its fluctuation. $V_{min/max}$ sets the cut-off voltage for low/high battery voltage condition. $[T_{min}, T_{max}]$ is the desired battery working temperature domain. SOC_{init} and SOC_{final} set the starting point and desired ending point for battery charging, respectively. The specific values will be shown in Section 4.3.4.

4.3.3 DeePC with Dimension Reduction

Even if the promising performance of (regularized) DeePC has been demonstrated in lots of applications, its time complexity is still high by solving the optimization problem with a large dimension of optimization variables $g \in \mathbb{R}^{T-T_{ini}-N+1}$, where T is often much larger than $T_{ini} + N$ to satisfy the persistent excitation (PE) condition. As a preliminary study, we exploited singular value decomposition (SVD) [113] for dimension reduction to facilitate the fast optimization while retaining the performance.

We denote the left-side data matrix of (4.2) as A and the right-side data vector of (4.2) as b :

$$A := \begin{bmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{bmatrix}, \quad b := \begin{bmatrix} u_{ini} \\ y_{ini} \\ u \\ y \end{bmatrix}. \tag{4.8}$$

Note that in order to satisfy the PE condition and attain good performance, a large number of columns in A are generally used [3], [90]. Since the fundamental principle is that the

columns of A spans the $T_{ini} + N$ input/output trajectories, we employed SVD to identify the l most important representative trajectories, which we refer to as *eigen-trajectories*. More specifically, we perform the following dimension reduction steps:

1. Let $p = (m + r)(T_{ini} + N)$, $q = T - T_{ini} - N + 1$ and $q \gg p$ for PE condition, then decompose the matrix A into:

$$A = W\Sigma V^\top, \quad (4.9)$$

where matrix $\Sigma \in \mathbb{R}^{p \times q}$ is a rectangular diagonal matrix and uniquely determined by A with all the singular values $\sigma_i = \Sigma_{ii}, i \leq \min\{p, q\}$ on the diagonal in descending order. The columns of matrix $W \in \mathbb{R}^{p \times p}$ and the columns of matrix $V \in \mathbb{R}^{q \times q}$ are called left-singular vectors and right-singular vectors of A , respectively. They also form two sets of orthonormal bases.

2. Choose the first l columns from V to form a new matrix which is denoted as $V_{[1:l]}$, where $l \leq q$. As the columns of V represent the principle mixture of system modes [120, p. C. 1] evolving in time, the first l columns of V are the corresponding most representative trajectories, which we call *eigen-trajectories*, and they are essentially utilized for mapping.
3. Define a new vector $\bar{g}$ with the relationship to g as,

$$g = V_{[1:l]}\bar{g}, \quad (4.10)$$

where $\bar{g}$ has the dimension l . The matrix $V_{[1:l]}$ here, as mentioned, is a mapping from $\bar{g}$ to g .

4. The equation (4.2) in the DeePC problem can be replaced by

$$\bar{A}\bar{g} = b, \quad (4.11)$$

where $\bar{A} = W\Sigma V^\top V_{[1:l]} = AV_{[1:l]}$, and the optimization variable changes from g to $\bar{g}$ with a lower dimension. The same equation transformation and replacement can be applied to (4.4b).

Remark 4.3.2 (Selection of Column Length l). *Three methods can be used to choose l for truncation of right-singular matrix V to get $V_{[1,l]}$,*

- *Singular value spectrum analysis. Plot the singular value in a descending order, l can be picked by identifying the 'elbow' point with steepest slope or some other points with required energy percentage along the curve.*
- *Cross-validation. it's a kind trial-and-error method by evaluating a performance index defined based on the variation of the length l .*
- *Optimal Hard Threshold Method [121]. An optimal singular value truncation point can be calculated by finding the underlying lower rank matrix, especially in the cases under heavy noise corruption.*

4.3.4 Simulations

In this section, the DeePC approach for Li-ion battery fast charging problem with safety constraints is implemented via simulation to demonstrate its effectiveness. In addition, comparison is also made for the proposed SVD-based dimension reduction strategy on DeePC, showing its advantages of promoting online optimization efficiency and, meanwhile, still retaining optimal performance of DeePC without dimension reduction.

4.3.4.1 Experimental Setup

A high fidelity battery simulation model - LIONSIMBA [122], is considered as the plant. This simulator is built based on the first-principle method and can provide all battery dynamics related signals, so the aforementioned signals, i.e. battery voltage, temperature, SOC and charging current, are measurable or estimated as those from a real Li-ion battery management system.

The offline data (u^d, y^d) are collected with the sampling interval chosen as $\Delta t = 10s$. The 'past' data length and 'future' data length are $T_{ini} = 60$ and $N = 70$, respectively. In the

Table 4.1 Safety constraint values.

I_{min}	ΔI_{min}	V_{min} [V]	T_{min} [$^{\circ}C$]	SOC_{init} [%]
0	$-0.1667C$	2.1	25	5
I_{max}	ΔI_{max}	V_{max} [V]	T_{max} [$^{\circ}C$]	SOC_{final} [%]
$2C$	$0.1667C$	4.17	31	95

cost function of DeePC problem, only one of the three outputs is penalized, i.e. SOC, with the corresponding weighting factor in $Q \in \mathbb{R}^{3 \times 3}$ is 10, and the weighting factor for charging current is $R = 0.1$. The coefficients of regularization terms are set as $\lambda_g = 10^5$, $\lambda_{y_{soc}, y_V} = 10^3$ and $\lambda_{y_T} = 10^6$. The lower and upper bounds for system constraints are listed in Table 4.1. Simulations are conducted using Matlab R2021b on Windows 10@3.6GHz PC with 8GB RAM. Before every simulation run, an ambient temperature $T_{amb} = 25^{\circ}C$ is set as the initial condition of simulator and the charging current $I = 0C$. Note that, for the chemistry property of this simulator, $1C$ value is approximate $30A/m^2$.

4.3.4.2 Performance of DeePC for Battery Fast Charging

In this simulation, a fast charging target is set to charge the Li-ion battery from SOC 5% to 95%, almost a full cycle charge of the Li-ion battery. The CCCV charging protocol is also implemented as the comparison set to demonstrate the effectiveness of DeePC for battery fast charging with safety constraints satisfaction. As shown in Fig. 4.2, three CCCV protocols are applied for Li-ion battery charging, where the pre-set cut-off voltage value is $V_{cutoff} = 4.15V$ and the constant charging currents are taken as $1.2C$, $1.5C$ and $2.0C$, respectively. So in this scheme, the battery is initially charged in CC mode before the battery voltage reaches the cut-off value, then CV mode begins for the rest of charging work until SOC equals 95%. As CCCV cannot handle system constraints, so the constraint violations of CCCV- $1.5C$ and CCCV- $2.0C$ protocols are depicted in Fig. 4.2b. Their temperatures rise above the safe zone during battery charging, it is hazardous to apply such charging protocols. To potentially overcome this problem, the CCCV schemes are often conservatively designed at the sacrifice

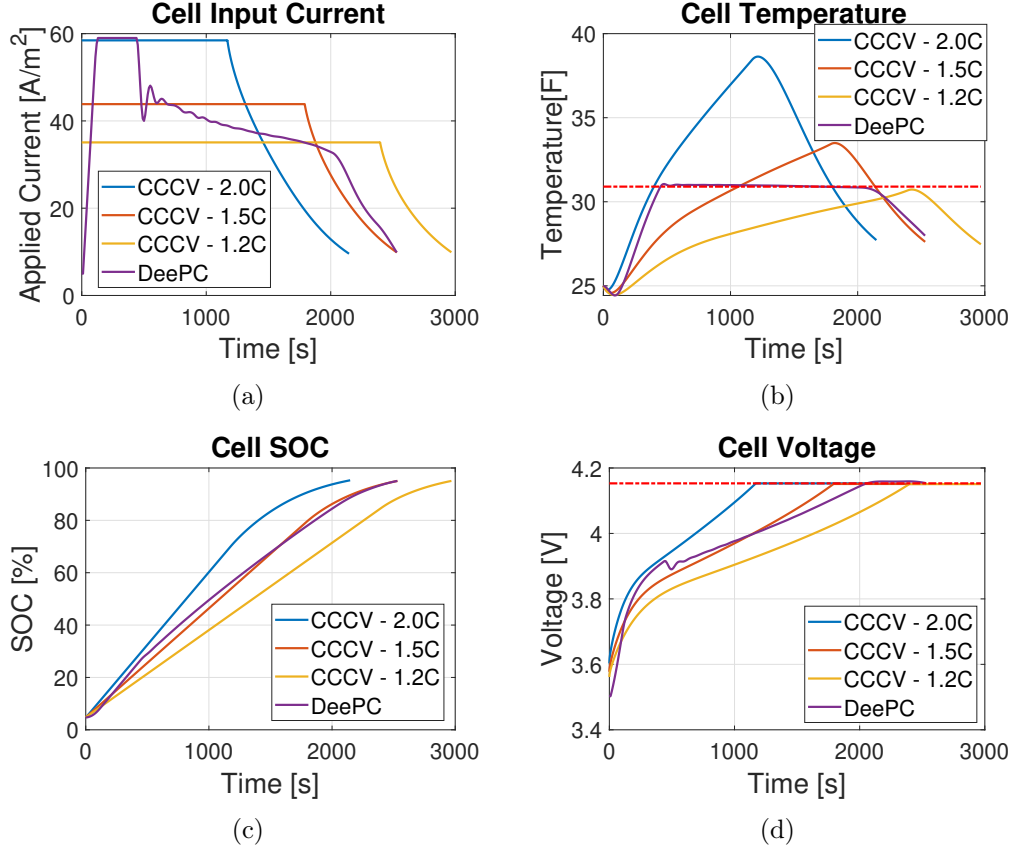


Figure 4.2 Simulation results of DeePC and CCCVs.

of charging speed. An typical example is the CCCV-1.2C charging protocol, it is carefully designed to just satisfy all constraints, but the charging speed is the slowest, SOC takes 2970s to rise from 5% to 95%.

DeePC is an optimal controller with the ability to deal with system constraints, it not only satisfies all safety constraints, but also has a relatively smooth charging current profile. The charging speed (2530s) of DeePC is almost the same as CCCV-1.5C (2531s) and has an approximate 15% increase when compared with CCCV-1.2C. In addition, to illustrate the prediction accuracy of DeePC for battery fast charging, its prediction performances are compared with real system measurements at the same time instant. As shown in Fig. 4.3a, the 1-step prediction performances of the three outputs match well with the system measurements. For the 10-step prediction case, Fig. 4.3b, even if certain deviations between successive prediction trajectories and the real measured ones, the overall trends of DeePC

output prediction still have good match to measurements. As a result, DeePC indeed has a good performance for Li-ion battery fast and safe charging with simple structure.

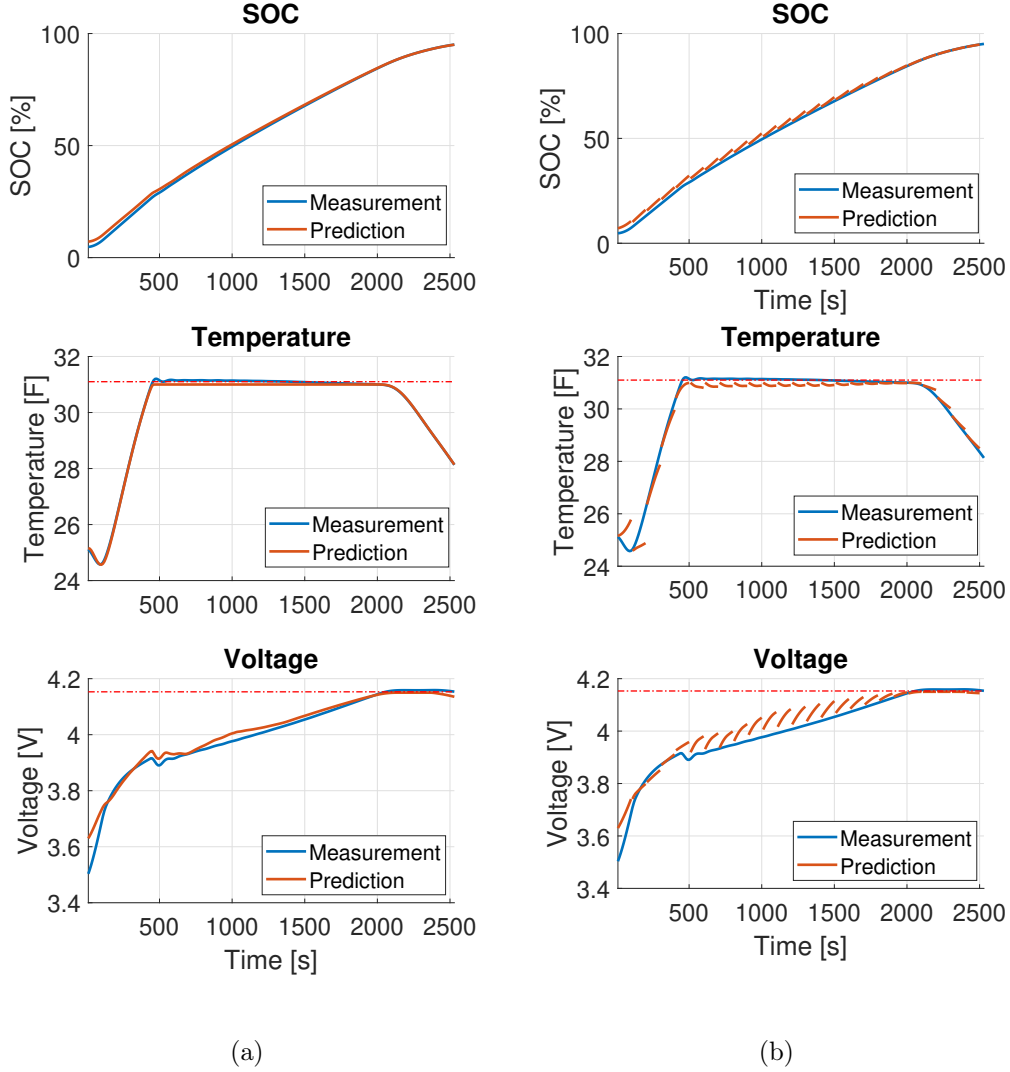


Figure 4.3 DeePC prediction vs real system measurement. (a). Comparison between $y_{1|t}$ and $y(t+1), t = 0, 1, 2, \dots$. (b). Comparison between $[y_{1|\bar{t}}, \dots, y_{10|\bar{t}}], \bar{t} = 0, 10, 20, \dots$ and $y(t+1), t = 0, 1, 2, \dots$.

4.3.4.3 Performance of DeePC with Dimension Reduction

The SVD-based dimension reduction method is applied in DeePC problem formulation, the simulation results in this section will show its ability to retain optimal control performance

while the computational burden is reduced a lot. Same conditions are set to initialize the Li-ion battery simulator. According to the steps proposed in section 4.3.3, after decomposition of matrix A , a truncated right-singular matrix $V_{[1:l]}$ is chosen with $l = 700$ (the total column length is $l_{max} = 1791$). Figure 4.4 shows the performance comparison between original DeePC and SVD-based efficient DeePC in the case of battery charging from SOC 49% to 80%. The closely overlapped system input and outputs directly demonstrate that the performance of proposed efficient DeePC can be as optimal as that of the original DeePC. Moreover, the costs of both DeePCs are basically the same, while the time-per-step optimization of the proposed efficient DeePC is much less. The corresponding results are summarized in Table 4.2. The average optimization time of efficient DeePC approach is above 4 times less than that of original DeePC, however, its performance and cost are barely impacted by the optimization variables with reduced dimension.

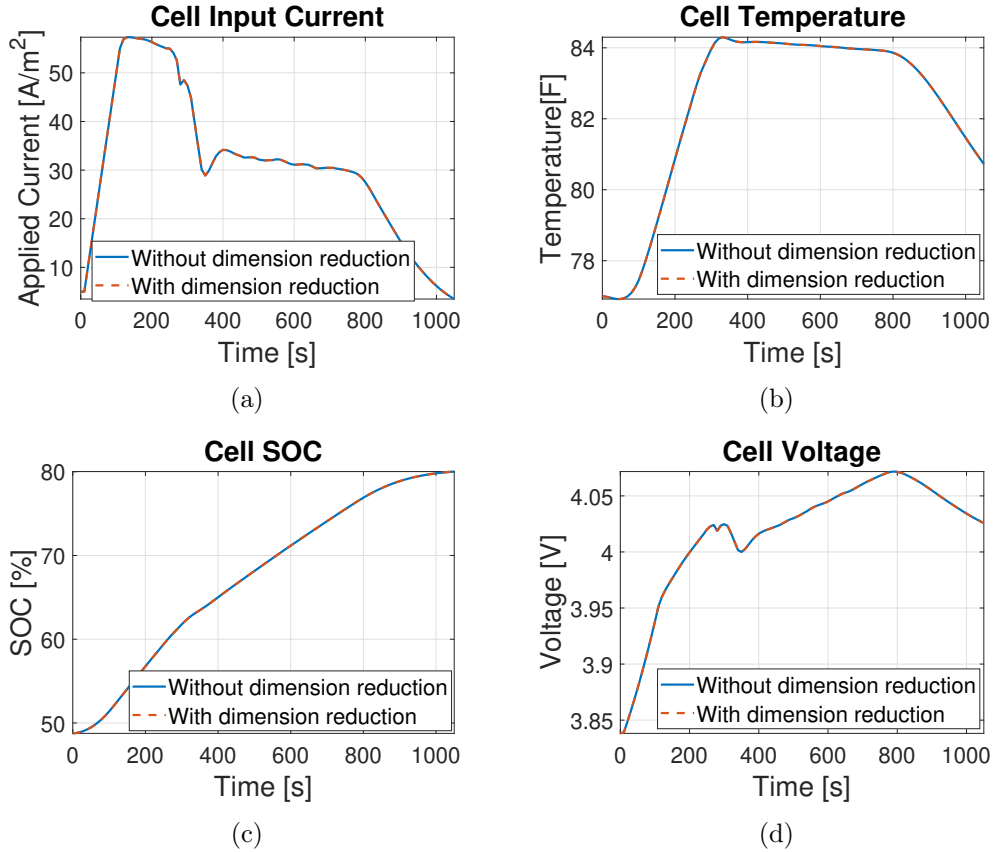


Figure 4.4 Simulation results of original DeePC and SVD-based DeePC.

Table 4.2 DeePC performance index based on A vs. $\bar{A}$.

	$A \in \mathbb{R}^{520 \times 1791}$	$A \in \mathbb{R}^{520 \times l} (l = 700)$
Cost	261813.25	261813.26
Time per Step [sec]	2.2s	0.5s

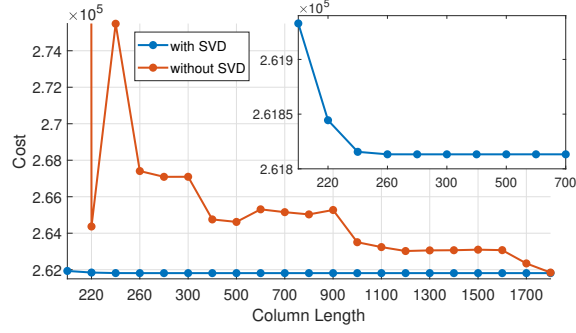


Figure 4.5 Cost comparison between SVD-based DeePC and original DeePC in the case of same Hankel matrix size.

Further simulations are run by varying l to form matrix $\bar{A}$ with different column lengths, so the dimension of the optimization variables is correspondingly varying. A comparison is conducted by directly truncating matrix A to make its size exactly the same as that of $\bar{A}$ so that the computational resource used by these two cases are also the same, and the results are shown in Fig. 4.5. With the reduction of column length from $l_{max} = 1791$, the cost function value of the optimization problem with SVD approach can still maintain stable and almost the same within a large range, i.e. $l \in [240, 1791]$, details are shown in the zoom-in window. However, DeePC without SVD approach (direct truncation of matrix A) shows an overall increasing cost by reducing columns of matrix A . According to the results shown in Fig. 4.5, if l is chosen to be 240, the computational time can further be reduced to 0.17s compared to the previous case with $l = 700$, and the cost is 261816, only increasing a little. So the SVD-based dimension reduction strategy can indeed reduce the computational complexity while retaining the optimal performance and cost as original DeePC scheme.

4.3.5 Conclusion

This chapter has presented the DeePC approach for Li-ion battery fast charging problem with safety consideration. The regularization strategy fits the DeePC scheme to nonlinear system. Only input/output data are needed for direct controller design without the intermediate parametric model identification process as those for EMs and EECMs. So the modeling effort and problem complexity are reduced a lot. In addition, to make DeePC online applicable for this fast charging task, a SVD-based dimension reduction approach is utilized to increase optimization speed. By properly choosing the eigen-trajectory matrix, dimension reduced optimization variables can still provide an optimal control. The simulation results confirm that by using the proposed method, the fast charging goal of Li-ion battery can be achieved, while avoiding unsafe charging conditions.

4.4 Privacy-preservation based DeePC for Automated Vehicle Control in Mixed Traffic Environment

4.4.1 Introduction of DeePC for Automated Vehicle Control in Mixed Traffic Environment

Advances in wireless communication technologies such as vehicle-to-infrastructure (V2I) or vehicle-to-vehicle (V2V) offer modern vehicles with enhanced connectivity and new opportunities for intelligent and integrated vehicle control [123], [124]. One typical example is the cooperative adaptive cruise control (CACC), which regulates a convoy of vehicles to improve traffic flow stability, safety, and energy efficiency [124]–[131]. While attracting much research interest, existing works on CACC predominately focus on effective platooning of pure connected and automated vehicles (CAVs). However, due to the gradual deployment of CAVs, it is expected that human-driven vehicles (HDVs) and CAVs will coexist for a long period of time, which raises the demand of developing cooperative control of CAVs in a mixed traffic flow.

HDVs often exhibit stochastic, uncertain behaviors that are difficult to characterize, mak-

ing it challenging for the control design of CAVs in a mixed traffic environment. To address this issue, several approaches have been proposed. For instance, in [132]–[134], feedback controllers of CAVs are designed under the formation of connected cruise control, where CAVs can receive information from HDVs ahead. Leading cruise control (LCC) is proposed in [135], where CAVs make decision by utilizing both the preceding and following HDVs’ information. To handle the prediction uncertainty of HDVs, a robust platoon control framework is designed in [136] by leveraging tube model predictive control (MPC). Note that existing works mainly focus on designing model-based approaches for CAVs [132]–[138]. One common strategy is to utilize car-following models to characterize the behavior of human drivers, e.g., optimal velocity model (OVM) [139] or intelligent driver model [140], enabling a state-space model of the mixed traffic system for system analysis and control design. Although model-based approaches can provide rigorous theoretical analysis and control synthesis when an accurate traffic model is available, it might not be applicable to real-world deployment since the parameters in human car-following models are non-trivial to identify accurately.

Instead of relying on explicit system models, data-driven approaches have recently emerged as an alternative to avoid model/parameter identification and directly incorporate collected data for control designs. For example, reinforcement learning (RL) [141]–[143] and adaptive dynamic programming (ADP) [144]–[146] have been developed to design optimal control schemes for CAVs. Although RL-based techniques can use experimental data to learn a model to simulate uncertain HDV behaviors, they are typically data intensive and have limited interpretability. On the other hand, data-driven ADP algorithms can provide optimal control actions with rigorous stability guarantees, but they struggle to handle constraints that are critical to vehicle safety. Recent advances in data-driven MPC have shown promise to achieve optimal control with constraint and stability guarantees [91], [147], [148]. In [90], [149], a Data-Enabled Predictive Leading Cruise Control (*DeeP-LCC*) strategy is developed for a mixed traffic system, which can efficiently handle safety constraints among multiple CAVs and HDVs. Specifically, by leveraging the Data-Enabled Predictive Control (*DeePC*)

techniques [3], input/output measurements are collected to first represent the mixed traffic system in a non-parametric manner, and to subsequently formulate a constrained optimization problem [90]. *DeePC* exploits behavioral system theory [150] and Willems’ fundamental lemma [69] to implicitly describe the system trajectories without explicitly carrying out model identification. Its receding horizon implementation is shown to be equivalent to the MPC formulation for linear time-invariant (LTI) systems, and it has found various successes in several practical applications [89], [92].

Employing the *DeeP-LCC* approach for the control of CAVs in the mixed traffic, however, poses great concerns in privacy. To enable coordinated control, the onboard data of vehicles, which may contain private information, needs to be extensively collected and shared via wireless communications (e.g., V2I or V2V), causing potential privacy leakage. Specifically, in a typical control architecture for a mixed-traffic fleet, each vehicle (both HDVs and CAVs) first sends its measured/estimated states to a central unit (e.g., a road-side edge or a remote cloud). The central unit then solves a pre-specified mixed-traffic optimal control problem and sends optimal control actions back to CAVs. In this setup, system measurements and control actions of CAVs need to be transmitted between the central unit and the local vehicles, raising concerns that an external eavesdropper or a honest-but-curious central unit can wiretap the communication channels to get sensitive information. In fact, several studies have shown that exposing local vehicle’s information to connectivity can lead to security vulnerabilities and various malicious activities [151]–[153]. Fail to protect privacy can potentially lead to devastating effects for CAVs and other vehicles sharing the roadway. Furthermore, it is worth noting that conventional privacy mechanisms either trade accuracy for privacy (e.g., differential privacy [74]) or incur heavy computation/communication overhead (e.g., crypton [154], [155]), and hence are inappropriate for mixed traffic system subject to stringent accuracy and real-time constraints.

Considering the aforementioned concerns and the growing awareness of security in cyber-physical systems, it is imperative to protect the privacy of CAVs in mixed traffic control. In

this chapter, we develop the first privacy-preserving data-enabled predictive control scheme for controlling CAVs in mixed traffic. Specifically, we consider the same mixed traffic system in [90], where multiple CAVs and HDVs cooperate with each other under the LCC framework [135]. We show that if the central unit is an honest-but-curious adversary or there exists an external eavesdropper, the *DeeP-LCC* under both Hankel and Page matrix structures cannot protect the private information of the vehicles. To avoid leaking the state and input information of the CAVs, a simple yet effective affine masking-based privacy protection method is designed, which can mask the true state and input signals. After the affine masking, an extended *DeeP-LCC* is derived to generate safe and optimal control actions for the CAVs. We further introduce a privacy notion and show that the proposed affine masking-based method can protect the private system state and input signals from being inferred by the attackers. Some preliminary results are summarized in a conference version [156]. The new contributions of this work are as follows.

First, the non-parametric representations of the mixed traffic system under Hankel and Page matrix structures are both studied. The previous *DeeP-LCC* scheme [90] leverages Hankel matrix to store the collected data for non-parametric system representation. We extend the *DeeP-LCC* with the Page matrix which is another effective structure for time-series data arrangement [157]. Second, we propose a privacy-preserving *DeeP-LCC* method for mixed traffic control, which can retain the advances of original *DeeP-LCC* while avoiding privacy leakage. We conceal the privacy-sensitive state and input signals of CAVs via affine masking and reformulate a compatible *DeeP-LCC* that is equivalent to the original one. Although the affine masking has been proved powerful to achieve privacy preservation in cloud-based MPC [158], [159], it has not been discussed for data-driven approaches. In this research, we successfully incorporate it with data-driven predictive control to enhance the privacy protection of the mixed traffic system. Furthermore, the affine masking-based method is light-weight in computation, making it extremely applicable to CAVs' control. Finally, extensive traffic simulations are conducted to validate the performance of the privacy-

preserving *DeeP-LCC*, which confirms the benefits of the proposed method in terms of not only improving traffic smoothness and fuel economy, but also protecting privacy against the attacker.

The remainder of this chapter is organized as follows. Section 4.4.2 formulates the mixed traffic control problem, and Section 4.4.3 provides an overview of *DeeP-LCC* method from [90]. Section 4.4.4 presents the affine masking-based privacy protection method and the extended *DeeP-LCC*. Section 4.4.5 analyses the equivalence and privacy-preserving properties of the developed method. Simulations are shown in Section 4.4.6. Finally, we conclude the chapter in Section 4.4.7.

4.4.2 System Model of Mixed Traffic

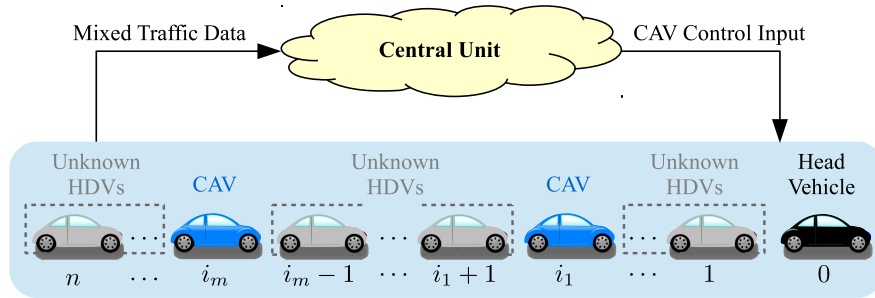


Figure 4.6 Schematic of mixed traffic system with $n + 1$ vehicles.

As shown in Fig. 4.6, the considered mixed traffic consists of $n + 1$ vehicles: one head vehicle (indexed as 0), m CAVs and $n - m$ HDVs. Let $\Omega = \{1, 2, \dots, n\}$ be the set of vehicle indices ordered from front to end. The sets of CAV indices and HDV indices are denoted by $\Omega_C = \{i_1, i_2, \dots, i_m\} \subseteq \Omega$ and $\Omega_H = \{j_1, j_2, \dots, j_{n-m}\} = \Omega \setminus \Omega_C$, respectively, where $i_1 < i_2 < \dots < i_m$ and $j_1 < j_2 < \dots < j_{n-m}$. We use $p_i(t)$, $v_i(t)$ and $a_i(t)$ to denote the position, velocity and acceleration of the i -th vehicle at time t , respectively.

The car-following dynamics for HDVs are modeled by the following nonlinear processes:

$$\dot{v}_i(t) = F(s_i(t), \dot{s}_i(t), v_i(t)), \quad i \in \Omega_H, \quad (4.12)$$

where $s_i(t) = p_{i-1}(t) - p_i(t)$ and $\dot{s}_i(t) = v_{i-1}(t) - v_i(t)$ are the relative spacing and velocity between vehicle i and its preceding vehicle $i - 1$, respectively. Denote s^* and v^* as the equilibrium spacing and equilibrium velocity of the mixed traffic system. Then the spacing error and velocity error of vehicle i are defined as $\tilde{s}_i(t) = s_i(t) - s^*$ and $\tilde{v}_i(t) = v_i(t) - v^*$, respectively. By applying the first-order Taylor expansion on (4.12) at the equilibrium (s^*, v^*) , the linearized model for each HDV can be obtained

$$\begin{cases} \dot{\tilde{s}}_i(t) = \tilde{v}_{i-1}(t) - \tilde{v}_i(t), \\ \dot{\tilde{v}}_i(t) = \alpha_1 \tilde{s}_i(t) - \alpha_2 \tilde{v}_i(t) + \alpha_3 \tilde{v}_{i-1}(t), \end{cases} \quad i \in \Omega_H, \quad (4.13)$$

with $\alpha_1 = \frac{\partial F}{\partial s}, \alpha_2 = \frac{\partial F}{\partial s} - \frac{\partial F}{\partial v}, \alpha_3 = \frac{\partial F}{\partial s}$ evaluated at the equilibrium state (s^*, v^*) . In this research, we use the OVM [139] model to describe the HDVs' car-following dynamics in (4.12). More details about the linearized model of OVM model can be found in [137], [138]. For the CAV, the acceleration is used as the control input, i.e., $\dot{v}_i(t) = u_i(t)$, $i \in \Omega_C$, and its system model is described by

$$\begin{cases} \dot{\tilde{s}}_i(t) = \tilde{v}_{i-1}(t) - \tilde{v}_i(t), \\ \dot{\tilde{v}}_i(t) = u_i(t), \end{cases} \quad i \in \Omega_C. \quad (4.14)$$

The state of each vehicle is denoted by $x_i(t) = [\tilde{s}_i, \tilde{v}_i]^\top$, and the overall state of the mixed traffic system is defined as

$$x(t) = [x_1^\top(t), x_2^\top(t), \dots, x_n^\top(t)]^\top \in \mathbb{R}^{2n}.$$

Based on (4.13) and (4.14), the linearized state-space model for the mixed traffic system can be derived, as follows:

$$\dot{x}(t) = Ax(t) + Bu(t) + H\epsilon(t), \quad (4.15)$$

where $u(t) = [u_{i_1}(t), u_{i_2}(t), \dots, u_{i_m}(t)]^\top$ is the collective control input and $\epsilon(t) = \tilde{v}_0(t) =$

$v_0(t) - v^*$ is the velocity error of the head vehicle. The system matrices in (4.15) are

$$\begin{aligned}
A &= \begin{bmatrix} A_{1,1} & & & & \\ A_{2,2} & A_{2,1} & & & \\ & \ddots & \ddots & & \\ & & A_{n-1,2} & A_{n-1,1} & \\ & & & A_{n,2} & A_{n,1} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \\
B &= \begin{bmatrix} e_{2n}^{2i_1}, e_{2n}^{2i_2}, \dots, e_{2n}^{2i_m} \end{bmatrix} \in \mathbb{R}^{2n \times m}, \\
H &= \begin{bmatrix} 1, \alpha_3, 0_{2n-2}^\top \end{bmatrix}^\top \in \mathbb{R}^{2n},
\end{aligned}$$

where

$$\begin{cases} A_{i,1} = \begin{bmatrix} 0 & -1 \\ \alpha_1 & -\alpha_2 \end{bmatrix}, A_{i,2} = \begin{bmatrix} 0 & 1 \\ 0 & \alpha_3 \end{bmatrix}, i \in \Omega_H; \\ A_{i,1} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, A_{i,2} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, i \in \Omega_C. \end{cases}$$

As discussed in [90], the spacing error of HDVs (i.e., $\tilde{s}_i$ ($i \in \Omega_H$)) is impractical to be observed due to the unknown human driving behavior. Therefore, the output signal of the mixed traffic system is constructed as

$$y(t) = \begin{bmatrix} x_{i_1}^\top(t), \dots, x_{i_m}^\top(t), \tilde{v}_{j_1}(t), \dots, \tilde{v}_{j_{n-m}}(t) \end{bmatrix}^\top,$$

where $y(t) \in \mathbb{R}^{n+m}$ consists of all state measurements of the CAVs, i.e., $\tilde{s}_i, \tilde{v}_i$ ($i \in \Omega_C$), and the velocity error signal of the HDVs, i.e., $\tilde{v}_i$ ($i \in \Omega_H$). The output $y(t)$ can be related to the overall state $x(t)$ via

$$y(t) = Cx(t), \quad (4.16)$$

with

$$C = \begin{bmatrix} e_{2n}^{2i_1-1}, e_{2n}^{2i_1}, \dots, e_{2n}^{2i_m-1}, e_{2n}^{2i_m}, e_{2n}^{2j_1}, e_{2n}^{2j_2}, \dots, e_{2n}^{2j_{n-m}} \end{bmatrix}^\top$$

being the output matrix. Given the sampling interval $\Delta t > 0$, the continuous-time model in (4.15) and (4.16) can be transformed into the discrete-time

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k) + H_d \epsilon(k), \\ y(k) = C_d x(k), \end{cases} \quad (4.17)$$

where $A_d = e^{A\Delta t} \in \mathbb{R}^{2n \times 2n}$, $B_d = \int_0^{\Delta t} e^{At} B dt \in \mathbb{R}^{2n \times m}$, $H_d = \int_0^{\Delta t} e^{At} H dt \in \mathbb{R}^{2n}$, $C_d = C \in \mathbb{R}^{(n+m) \times 2n}$. Let $\hat{u}(k) = \begin{bmatrix} \epsilon(k), u^\top(k) \end{bmatrix}^\top$ be the combined inputs signal and $\hat{B}_d = \begin{bmatrix} H_d, B_d \end{bmatrix}$ be the combined input matrix. Then (4.17) can be rewritten into a compact form as follows:

$$\begin{cases} x(k+1) = A_d x(k) + \hat{B}_d \hat{u}(k), \\ y(k) = C_d x(k). \end{cases} \quad (4.18)$$

In this chapter, we assume that $\alpha_1 - \alpha_2 \alpha_3 + \alpha_3^2 \neq 0$, which ensures that the mixed traffic system is stabilizable and observable [135]. Note that the above model is only used for analysis, and the controller design in Section 4.4.3 does not assume the exact model (4.17).

4.4.3 Data-Enabled Predictive Leading Cruise Control

4.4.3.1 Non-Parametric Representation of Mixed Traffic

Conventional methods rely on explicit system model (4.17) to facilitate controller design. One typical example is MPC whose performance depends closely on the model accuracy. Even though there exist many system identification techniques, it is still difficult to obtain accurate models for complex systems such as the mixed traffic system with stochastic and uncertain human driving behavior. The *DeePC* [3] is a promising model-free paradigm and recently has been leveraged by [90] to design the *DeeP-LCC* scheme for mixed traffic. In particular, based on Willems' fundamental lemma [69], this class of data-driven approaches can describe the dynamical system in a non-parametric manner.

We now discuss the non-parametric representation of mixed traffic system (4.17). It begins with data collection. Specifically, a trajectory of length $T \in \mathbb{Z}_+$ is collected from the mixed traffic system (4.17), which includes the following two parts:

- a combined input sequence of the mixed traffic system

$$\hat{u}_{[1,T]}^d = \begin{bmatrix} \hat{u}^d(1) \\ \vdots \\ \hat{u}^d(T) \end{bmatrix} \in \mathbb{R}^{(m+1)T},$$

which is comprised of CAVs' acceleration sequence and the velocity error sequence of the head vehicle, i.e.,

$$u_{[1,T]}^d = \begin{bmatrix} u^d(1) \\ \vdots \\ u^d(T) \end{bmatrix} \in \mathbb{R}^{mT}, \epsilon_{[1,T]}^d = \begin{bmatrix} \epsilon^d(1) \\ \vdots \\ \epsilon^d(T) \end{bmatrix} \in \mathbb{R}^T;$$

- the corresponding output sequence

$$y_{[1,T]}^d = \begin{bmatrix} y^d(1) \\ \vdots \\ y^d(T) \end{bmatrix} \in \mathbb{R}^{(n+m)T}.$$

After collecting these data sequences, different matrix structures can be used to store them for non-parametric representation. The Hankel matrix is the commonly used one [3], [90]. Specifically, under the Hankel matrix structure, the data sequences are divided into two parts, i.e., the “previous data” of length $T_{\text{ini}} \in \mathbb{Z}_+$ and “future data” of length $N \in \mathbb{Z}_+$, to construct the following matrices:

$$\begin{bmatrix} U_p^{\mathcal{H}} \\ U_f^{\mathcal{H}} \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (u_{[1,T]}^d), \begin{bmatrix} E_p^{\mathcal{H}} \\ E_f^{\mathcal{H}} \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (\epsilon_{[1,T]}^d),$$

$$\begin{bmatrix} Y_p^{\mathcal{H}} \\ Y_f^{\mathcal{H}} \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (y_{[1,T]}^d),$$

where $U_p^{\mathcal{H}}$ denotes the first T_{ini} block rows of $\mathcal{H}_{T_{\text{ini}}+N}(u_{[1,T]}^d)$ and $U_f^{\mathcal{H}}$ denotes the last N block rows of $\mathcal{H}_{T_{\text{ini}}+N}(u_{[1,T]}^d)$, respectively (similarly for $E_p^{\mathcal{H}}, E_f^{\mathcal{H}}$ and $Y_p^{\mathcal{H}}, Y_f^{\mathcal{H}}$). Let $u_{\text{ini}} = u_{[t-T_{\text{ini}}, t-1]}$ be the control input sequence within a past time horizon of length T_{ini} , and $u = u_{[t, t+N-1]}$ be the control input sequence within a prediction horizon of length N (similarly for $\epsilon_{\text{ini}}, \epsilon$ and y_{ini}, y). Based on Willems' fundamental lemma [69] and the *DeePC* [3], we have the following proposition.

Proposition 4.4.1. *Suppose the combined input sequence $\hat{u}_{[1,T]}^d$ is Hankel exciting of order $T_{\text{ini}} + N + 2n$. Then a trajectory $(u_{\text{ini}}, \epsilon_{\text{ini}}, y_{\text{ini}}, u, \epsilon, y)$ belongs to the mixed traffic system (4.17) if there exists $g \in \mathbb{R}^{T-T_{\text{ini}}-N+1}$ such that*

$$\begin{bmatrix} U_p^{\mathcal{H}} \\ E_p^{\mathcal{H}} \\ Y_p^{\mathcal{H}} \\ U_f^{\mathcal{H}} \\ E_f^{\mathcal{H}} \\ Y_f^{\mathcal{H}} \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ \epsilon_{\text{ini}} \\ y_{\text{ini}} \\ u \\ \epsilon \\ y \end{bmatrix}. \quad (4.19)$$

4.4.3.2 *DeeP-LCC* Formulation

Proposition 4.4.1 reveals that if sufficiently rich traffic data is collected, the future trajectory of the mixed traffic system can be directly predicted without relying on explicit system model. The relations (4.19) is the non-parametric representations of the mixed traffic system, which are the key in formulating the *DeeP-LCC*. More precisely, at each time step t , the control actions of the CAVs are generated by solving the following optimization problem:

$$\begin{aligned}
\min_{g,u,y} \quad & J(y, u) = \sum_{k=t}^{t+N-1} \left(\|y(k)\|_Q^2 + \|u(k)\|_R^2 \right) \\
\text{s.t.} \quad & (4.19) \\
& \epsilon = 0_N, \\
& y_{\min} \leq y(k) \leq y_{\max}, \quad k = t, \dots, t + N - 1, \\
& a_{\min} \leq u(k) \leq a_{\max}, \quad k = t, \dots, t + N - 1.
\end{aligned} \tag{4.20}$$

In (4.20), $J(y, u)$ is a quadratic cost function, and its weighting matrices Q and R are selected as $Q = \text{diag}(Q_C, Q_H) \in \mathbb{R}^{(n+m) \times (n+m)}$ with $Q_C = \text{diag}(w_s, w_v, \dots, w_s, w_v) \in \mathbb{R}^{2m \times 2m}$, $Q_H = \text{diag}(w_v, \dots, w_v) \in \mathbb{R}^{(n-m) \times (n-m)}$ and $R = \text{diag}(w_u, \dots, w_u) \in \mathbb{R}^{m \times m}$, where w_s , w_v , w_u are the weighting factors for the spacing error of CAVs, the velocity error of all vehicles, and the control input of CAVs, respectively. The second constraint, $\epsilon = 0_N$, is used to predict the future velocity error sequence of the head vehicle. The third and fourth constraints are applied to the output and input of the mixed traffic system for safety guarantees, where $y_{\min}$ and $y_{\max}$ denote the lower and upper bounds of output signal, respectively (similarly for $a_{\min}$, $a_{\max}$). We refer the interested reader to [90] for more details on designing the cost function and constraints in *DeeP-LCC*.

4.4.4 Privacy-Preserving *DeeP-LCC*

In this section, we first introduce the attack models, and then present an affine masking strategy. Finally, a new privacy-preserving *DeeP-LCC* is proposed.

4.4.4.1 Attack Model

For the mixed-traffic vehicle fleet discussed above, a central unit (e.g., a road-side edge compute or a remote cloud) is needed to receive all the vehicle data and solve the optimization problem (4.20). A feasible architecture is shown in Fig. 4.7 and is described as follows:

- *Handshaking Phase*: The vehicle system sends $\left\{ Q, R, (u_{[1,T]}^d, \epsilon_{[1,T]}^d, y_{[1,T]}^d), (y_{\min}, y_{\max}, \right.$

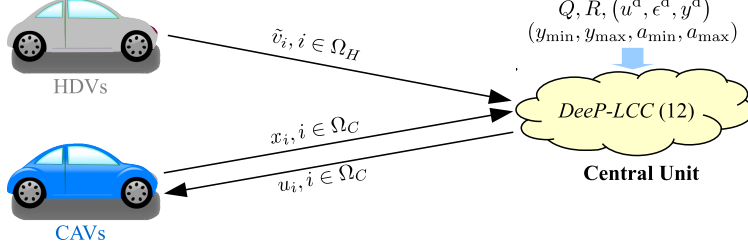


Figure 4.7 Unsecure *DeeP-LCC* architecture.

$a_{\min}, a_{\max})$ to the central unit, which are the necessary information for the central unit to set up the optimization problem (4.20).

- *Execution Phase*: At each time step k , each CAV and HDV sends its state $x_i(k) = \begin{bmatrix} \tilde{s}_i(k), \tilde{v}_i(k) \end{bmatrix}^\top$ ($i \in \Omega_C$) and velocity error $\tilde{v}_i(k)$ ($i \in \Omega_H$), respectively, to the central unit. The central unit computes $u(k)$ by solving the optimization problem (4.20) and sends optimal $u_i(k)$ ($i \in \Omega_C$) to the CAVs. Finally, the CAVs apply $u_i(k)$ to the actuators and the system evolves over one step.

The involved vehicles need to provide the central unit with real-time measurements, pre-collected data, cost function, and constraints to facilitate the calculation of (4.20). The data may contain private contents that need to be protected from external attackers. In this research, we consider two attack models [160]:

- *Eavesdropping attacks* are attacks in which an external eavesdropper wiretaps communication channels to intercept exchanged messages in an attempt to learn the information about sending parties.
- *Honest-but-curious attacks* are attacks in which the untrusted central unit follows the protocol steps correctly but is curious and collects received intermediate data in an attempt to learn the information about the vehicles.

In particular, we consider the case that the privacy-sensitive information is contained in the system state and input of CAVs, i.e., $x_i(k) = \begin{bmatrix} \tilde{s}_i(k), \tilde{v}_i(k) \end{bmatrix}^\top, u_i(k), i \in \Omega_C$. It is

apparent that the attacker can successfully eavesdrop the messages $x_i(k)$ and $u_i(k)$ when the *DeeP-LCC* architecture introduced in Section 4.4.3.2 is adopted.

In the next subsection, we develop a masking mechanism to modify the exchanged information between the CAVs and the central unit such that an equivalent data-enabled predictive control problem is solved without affecting system performance while preventing the attacker from wiretapping the CAVs' state and input. Although we only consider the privacy preservation for CAVs, the following proposed approach can also be straightforwardly adopted by HDVs to mask privacy-sensitive information, e.g., velocity error $\tilde{v}_i(k)$, $i \in \Omega_H$.

4.4.4.2 Affine Masking

Affine masking is a class of algebraic transformations and recently has been adopted to accomplish privacy protection for cloud-based control [158], [159]. In this research, by considering the characteristics of the mixed traffic system, we introduce affine transformation maps to design a privacy-preserved *DeeP-LCC* scheme. Specifically, for each CAV, two invertible affine maps are employed to transform the state $x_i(k)$ and input $u_i(k)$ to the new state $\bar{x}_i(k)$ and input $\bar{u}_i(k)$, as follows:

$$\begin{cases} \bar{x}_i(k) = P_{x,i}x_i(k) + l_{x,i}, \\ \bar{u}_i(k) = P_{u,i}u_i(k) + l_{u,i}, \end{cases} \quad \forall i \in \Omega_C, \quad (4.21)$$

where $P_{x,i} \in \mathbb{R}^{2 \times 2}$ is an arbitrary invertible matrix, and $l_{x,i} \in \mathbb{R}^2$, $P_{u,i} \in \mathbb{R}$ and $l_{u,i} \in \mathbb{R}$ are arbitrary non-zero vector or constant with compatible dimensions. After each CAV masks its true state and input via (4.21), the new input and output signals of the mixed traffic system are defined as

$$\begin{aligned} \bar{u}(k) &= \left[\bar{u}_{i_1}(k), \bar{u}_{i_2}(k), \dots, \bar{u}_{i_m}(k) \right]^\top, \\ \bar{y}(k) &= \left[\bar{x}_{i_1}^\top(k), \dots, \bar{x}_{i_m}^\top(k), \tilde{v}_{j_1}(k), \dots, \tilde{v}_{j_{n-m}}(k) \right]^\top. \end{aligned} \quad (4.22)$$

Based on (4.21), (4.22) and the definition of $u(k)$ and $y(k)$, we obtain that

$$\bar{u}(k) = P_u u(k) + L_u, \quad \bar{y}(k) = P_y y(k) + L_y, \quad (4.23)$$

where

$$\begin{aligned} P_u &= \text{diag}(P_{u,i_1}, P_{u,i_2}, \dots, P_{u,i_m}) \in \mathbb{R}^{m \times m}, \\ L_u &= \begin{bmatrix} l_{u,i_1}, l_{u,i_2}, \dots, l_{u,i_m} \end{bmatrix}^\top \in \mathbb{R}^m, \\ P_y &= \text{diag}(P_{x,i_1}, \dots, P_{x,i_m}, I_{n-m}) \in \mathbb{R}^{(n+m) \times (n+m)}, \\ L_y &= \begin{bmatrix} l_{x,i_1}^\top, \dots, l_{x,i_m}^\top, 0_{n-m}^\top \end{bmatrix}^\top \in \mathbb{R}^{(n+m)}. \end{aligned}$$

In (4.23), (P_u, L_u) and (P_y, L_y) are two affine maps constructed from the local affine maps of CAVs, and are used to transform the original input and output of mixed traffic system, i.e., $(u(k), y(k))$, into the new ones $(\bar{u}(k), \bar{y}(k))$. Given the affine transformation, the discrete-time model of the mixed traffic system (4.17) can be reformulated as follows:

$$\begin{cases} x(k+1) = A_d x(k) + \bar{B}_d \bar{u}(k) + H_d \epsilon(k) + \bar{L}_u, \\ \bar{y}(k) = \bar{C}_d x(k) + L_y, \end{cases} \quad (4.24)$$

where $\bar{B}_d = B_d P_u^{-1}$, $\bar{L}_u = -B_d P_u^{-1} L_u$, and $\bar{C}_d = P_y C_d$.

Under the affine transformation mechanism (4.22), the exchanged information between the CAVs and the central unit changes to $\bar{x}_i(k), \bar{u}_i(k), i \in \Omega_C$, which can avoid leaking the real state and input signals to the eavesdropper or the central unit. This affine transformation mechanism also leads to a new system formulation (4.24), and thus a compatible *DeeP-LCC* method needs to be developed.

4.4.4.3 Privacy-Preserving *DeeP-LCC* Reformulation

Denote $\bar{u}_{[1,T]}^d = \begin{bmatrix} \bar{u}^d(1), \dots, \bar{u}^d(T) \end{bmatrix}^\top$ as the corresponding acceleration sequence of $u_{[1,T]}^d$ under the affine map (P_u, L_u) , and $\bar{y}_{[1,T]}^d = \begin{bmatrix} \bar{y}^d(1), \dots, \bar{y}^d(T) \end{bmatrix}^\top$ as the corresponding output sequence of $y_{[1,T]}^d$ under the affine map (P_y, L_y) . Then, similar to $\hat{u}_{[1,T]}^d$, the combined input

sequence $\hat{u}_{[1,T]}^d \in \mathbb{R}^{(m+1)T}$ is constructed with $\bar{u}_{[1,T]}^d$ and $\epsilon_{[1,T]}^d$. The matrices $(\bar{U}_p^{\mathcal{H}}, \bar{U}_f^{\mathcal{H}}, \bar{Y}_p^{\mathcal{H}}, \bar{Y}_f^{\mathcal{H}})$ can be constructed with $\bar{u}_{[1,T]}^d$ and $\bar{y}_{[1,T]}^d$ by following the same procedure shown in (4.19). Motivated by Willems' fundamental lemma [69], (4.19), we use the data $(\bar{u}_{[1,T]}^d, \epsilon_{[1,T]}^d, \bar{y}_{[1,T]}^d)$ to represent the affine masking-based mixed traffic system (4.24) under the Hankel matrix structures. The results are summarized below.

Proposition 4.4.2. *Suppose the data $\hat{u}_{[1,T]}^d$ is Hankel exciting of order $T_{\text{ini}} + N + 2n + 1$. Then, $(\bar{u}_{\text{ini}}, \epsilon_{\text{ini}}, \bar{y}_{\text{ini}}, \bar{u}, \epsilon, \bar{y})$ is a trajectory of (4.24) if there exists $\bar{g} \in \mathbb{R}^{T-T_{\text{ini}}-N+1}$ such that*

$$\begin{bmatrix} \bar{U}_p^{\mathcal{H}} \\ E_p^{\mathcal{H}} \\ \bar{Y}_p^{\mathcal{H}} \\ \bar{U}_f^{\mathcal{H}} \\ E_f^{\mathcal{H}} \\ \bar{Y}_f^{\mathcal{H}} \\ 1_{T-T_{\text{ini}}-N+1}^\top \end{bmatrix} \bar{g} = \begin{bmatrix} \bar{u}_{\text{ini}} \\ \epsilon_{\text{ini}} \\ \bar{y}_{\text{ini}} \\ \bar{u} \\ \epsilon \\ \bar{y} \\ 1 \end{bmatrix}. \quad (4.25)$$

The affine maps are able to mask the true system state $x_i(k)$ and input $u_i(k)$ of CAVs to protect the privacy, and in the central unit, a new optimization problem with respect to $(\bar{y}, \bar{u})$ and the new non-parametric representation (4.25) are solved. Specifically, with the affine maps, one can show that (4.20) can be transformed into the following problem:

$$\begin{aligned} \min_{\bar{g}, \bar{u}, \bar{y}} \quad & \bar{J}(\bar{y}, \bar{u}) \\ \text{s.t.} \quad & (4.25) \\ & \epsilon = 0_N, \end{aligned} \quad (4.26)$$

$$\bar{y}_{\min} \leq \bar{y}(k) \leq \bar{y}_{\max}, \quad k = t, \dots, t + N - 1,$$

$$\bar{a}_{\min} \leq \bar{u}(k) \leq \bar{a}_{\max}, \quad k = t, \dots, t + N - 1,$$

where the cost function is defined as

$$\bar{J}(\bar{y}, \bar{u}) = \sum_{k=t}^{t+N-1} \left(\|\bar{y}(k)\|_{\bar{Q}}^2 + \bar{q}^\top \bar{y}(k) + \|\bar{u}(k)\|_{\bar{R}}^2 + \bar{r}^\top \bar{u}(k) \right)$$

with $\bar{Q} \in \mathbb{R}^{(n+m) \times (n+m)}$, $\bar{q} \in \mathbb{R}^n$, $\bar{R} \in \mathbb{R}^{m \times m}$, $\bar{r} \in \mathbb{R}^m$, $\bar{Q}_f \in \mathbb{R}^{n \times n}$, and $\bar{q}_f \in \mathbb{R}^n$ given by

$$\begin{aligned}\bar{Q} &= P_y^{-\top} Q P_y^{-1} \in \mathbb{R}^{(n+m) \times (n+m)}, \bar{q} = -2\bar{Q} L_y \in \mathbb{R}^{(n+m)}, \\ \bar{R} &= P_u^{-\top} R P_u^{-1} \in \mathbb{R}^{m \times m}, \bar{r} = -2\bar{R} L_u \in \mathbb{R}^m.\end{aligned}\tag{4.27}$$

In (4.26), $(\bar{y}_{\min}, \bar{y}_{\max})$ and $(\bar{u}_{\min}, \bar{u}_{\max})$ are the corresponding constraint bounds of $(y_{\min}, y_{\max})$ and $(u_{\min}, u_{\max})$ under the affine maps (P_y, L_y) and (P_u, L_u) given in (4.23), i.e., $\bar{y}_{\min} = P_y y_{\min} + L_y$, $\bar{y}_{\max} = P_y y_{\max} + L_y$ and $\bar{u}_{\min} = P_u y_{\min} + L_u$, $\bar{u}_{\max} = P_u y_{\max} + L_u$.

Compared to the unsecure *DeeP-LCC* in Section 4.4.4.1, our privacy-preserved *DeeP-LCC* architecture with the affine maps, shown in Fig. 4.8, is modified as:

- *Handshaking Phase:* The vehicle system sends

$$\{\bar{Q}, \bar{R}, \bar{q}, \bar{r}, (\bar{u}_{[1,T]}^d, \epsilon_{[1,T]}^d, \bar{y}_{[1,T]}^d), (\bar{y}_{\min}, \bar{y}_{\max}, \bar{a}_{\min}, \bar{a}_{\max})\}$$

to the central unit, that is, the necessary information to set up the optimization problem (4.26).

- *Execution Phase:* At each time step k , the CAVs encodes $x_i(k)$ ($i \in \Omega_C$) into $\bar{x}_i(k)$ with $(P_{x,i}, l_{x,i})$ and sends $\bar{x}_i(k)$ to the central unit. Meanwhile, HDVs sends velocity error $\tilde{v}_i(k)$ ($i \in \Omega_H$) to the central unit. After receiving these data, the central unit computes $\bar{u}(k)$ by solving the optimization problem (4.26) and sends the solution $\bar{u}_i(k)$ ($i \in \Omega_C$) to the CAVs. Finally, each CAV uses $(P_{u,i}, l_{u,i})$ to decode $\bar{u}_i(k)$, i.e., $u_i(k) = P_{u,i}^{-1} (\bar{u}_i(k) - l_{u,i})$ and applies $u_i(k)$ to the actuators. The system then evolves over one step.

Remark 4.4.3. The formulations (4.25) is valid for deterministic LTI system (4.24). However, in practice, the car-following behavior of HDVs is stochastic and has uncertainties, which leads to a non-deterministic and nonlinear mixed traffic system. Inspired by the regularization design for standard *DeePC* [3], slack variable $\bar{\sigma}_y \in \mathbb{R}^{(n+m)T_{\text{ini}}}$ and two-norm

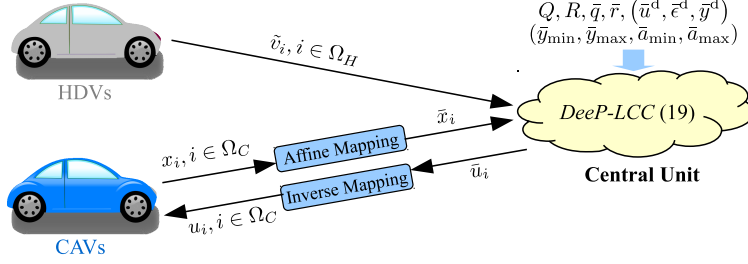


Figure 4.8 Privacy-preservation *DeeP-LCC* architecture.

regularization on $\bar{g}$ can be introduced to handle system uncertainties and nonlinearities. For instance, the optimization problem (4.26) under the Hankel matrix structure becomes

$$\begin{aligned}
 & \min_{\bar{g}, \bar{u}, \bar{y}} \quad \bar{J}(\bar{y}, \bar{u}) + \bar{\lambda}_g \|\bar{g}\|_2^2 + \bar{\lambda}_\sigma \|\bar{\sigma}_y\|_2^2 \\
 & \text{s.t.} \quad \begin{bmatrix} \bar{U}_p \\ E_p \\ \bar{Y}_p \\ \bar{U}_f \\ E_f \\ \bar{Y}_f \\ 1_{T-T_{\text{ini}}-N+1} \end{bmatrix} \bar{g} = \begin{bmatrix} \bar{u}_{\text{ini}} \\ \epsilon_{\text{ini}} \\ \bar{y}_{\text{ini}} \\ \bar{u} \\ \epsilon \\ \bar{y} \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \bar{\sigma}_y \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\
 & \quad \epsilon = 0_N, \\
 & \quad \bar{y}_{\min} \leq \bar{y}(k) \leq \bar{y}_{\max}, \quad k = t, \dots, t + N - 1, \\
 & \quad \bar{a}_{\min} \leq \bar{u}(k) \leq \bar{a}_{\max}, \quad k = t, \dots, t + N - 1,
 \end{aligned} \tag{4.28}$$

where $\bar{\lambda}_g > 0$ and $\bar{\lambda}_\sigma > 0$ are weighting coefficients. The slack variable $\bar{\sigma}_y$ for the past output signal is added to ensure feasibility of the equality constraint, while the regularization on $\bar{g}$ is used to avoid overfitting.

4.4.5 Equivalence and Privacy Preservation

As mentioned in Section 4.4.4.1, the attacker aims to infer the system state and input of CAVs, i.e., $x_i(k) = \begin{bmatrix} \tilde{s}_i(k), \tilde{v}_i(k) \end{bmatrix}^\top, u_i(k), i \in \Omega_C$. Note that under the privacy-preserving

DeeP-LCC architecture, the exchanged information between CAVs and the central unit during the execution phase is $\bar{x}_i(k)$ and $\bar{u}_i(k)$ rather than the actual system state and input.

In the section, we first prove that the reformulated *DeeP-LCC* problem (4.26) is equivalent to the original *DeeP-LCC* problem (4.20), and then we show that the privacy of CAVs' state and input is indeed protected.

4.4.5.1 Equivalence with Affine Transformation

The following Lemma establishes that the equivalence of the reformulated *DeeP-LCC* problem (4.26) to the original *DeeP-LCC* problem (4.20).

Lemma 4.4.4. *Under the affine transformation mechanism, the optimization problem (4.26) is equivalent to (4.20).*

Proof. With the input and output transformations in (4.23), the cost term transformations in (4.27), and the definitions of $J(\cdot, \cdot)$ and $\bar{J}(\cdot, \cdot)$, it can be shown that

$$J(y, u) = \bar{J}(\bar{y}, \bar{u}) + \varrho, \quad (4.29)$$

where $\varrho = \sum_{k=t}^{t+N-1} (L_y^\top \bar{Q} L_y + L_u^\top \bar{R} L_u) \in \mathbb{R}$ is a constant. We now use proof by contradiction. Assume that $(\bar{y}^*, \bar{u}^*)$ is the global minimizer of optimization problem (4.26), and (y^*, u^*) is the corresponding sequence of $(\bar{y}^*, \bar{u}^*)$ under the inverse affine maps (P_y, L_y) , (P_u, L_u) , i.e.,

$$\begin{aligned} y^* &= \begin{bmatrix} y^*(t) \\ \vdots \\ y^*(t+N-1) \end{bmatrix} = \begin{bmatrix} P_y^{-1}(y^*(t) - L_y) \\ \vdots \\ P_y^{-1}(y^*(t+N-1) - L_y) \end{bmatrix}, \\ u^* &= \begin{bmatrix} u^*(t) \\ \vdots \\ u^*(t+N-1) \end{bmatrix} = \begin{bmatrix} P_u^{-1}(u^*(t) - L_u) \\ \vdots \\ P_u^{-1}(u^*(t+N-1) - L_u) \end{bmatrix}. \end{aligned}$$

As $(\bar{y}^*, \bar{u}^*)$ is a trajectory of system (4.24) and satisfies the constraints in (4.26), it can be confirmed that (y^*, u^*) is a trajectory of system (4.17) and satisfies the constraints in (4.20).

We also assume that (y^*, u^*) is not the global minimizer of problem (4.20), and thus there exists an optimal sequence (y^{**}, u^{**}) (other than (y^*, u^*)) such that

$$J(y^{**}, u^{**}) < J(y^*, u^*). \quad (4.30)$$

Let $(\bar{y}^{**}, \bar{u}^{**})$ be the corresponding trajectory of (y^{**}, u^{**}) under the forward affine maps (P_y, L_y) , (P_u, L_u) . According to (4.29), (4.30) can be rewritten as

$$\bar{J}(\bar{y}^{**}, \bar{u}^{**}) + \varrho < \bar{J}(\bar{y}^*, \bar{u}^*) + \varrho, \quad (4.31)$$

which contradicts the assumption that $(\bar{y}^*, \bar{u}^*)$ is the global minimizer of optimization problem (4.26). Therefore, if $(\bar{y}^*, \bar{u}^*)$ is the global minimizer of problem (4.26), then its inverse affine transformation, i.e., (y^*, u^*) , should be solution to the original problem (4.20), indicating that these two problems are equivalent. $\square$

4.4.5.2 Privacy Preservation

We next discuss the privacy notion used in this research. As mentioned in Section 4.4.4.1, the attacker aims to infer the system state $x_i(k)$ and control input $u_i(k)$ of CAVs. Under the privacy-preserving architecture discussed above, the attacker will have access to $\bar{x}_i(k)$ and $\bar{u}_i(k)$ at each time step k , and we need to show that for any $\kappa \in \mathbb{Z}_+$, $x_{i,[1,\kappa]}$ and $u_{i,[1,\kappa]}$ cannot be identified from $\bar{x}_{i,[1,\kappa]}$ and $\bar{u}_{i,[1,\kappa]}$. According to (4.21), we use

$$(x_{i,[1,\kappa]}, u_{i,[1,\kappa]}) \xrightarrow{(P_{x,i}, l_{x,i}, P_{u,i}, l_{u,i})} (\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]}) , \quad \forall i \in \Omega_C$$

to denote that $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ is the transformed trajectory of $(x_{i,[1,\kappa]}, u_{i,[1,\kappa]})$ under the affine maps $(P_{x,i}, l_{x,i})$ and $(P_{u,i}, l_{u,i})$. For any feasible state sequence $\bar{x}_{i,[1,\kappa]}$ and input sequence $\bar{u}_{i,[1,\kappa]}$ received by the attacker, the set $\Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ is defined as

$$\begin{aligned} \Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]}) &= \{x_{i,[1,\kappa]}, u_{i,[1,\kappa]} : \exists (P_{x,i}, l_{x,i}, P_{u,i}, l_{u,i}) \\ \text{s.t. } (x_{i,[1,\kappa]}, u_{i,[1,\kappa]}) &\xrightarrow{(P_{x,i}, l_{x,i}, P_{u,i}, l_{u,i})} (\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})\}. \end{aligned}$$

Essentially, the set $\Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ includes all possible values of $(x_{i,[1,\kappa]}, u_{i,[1,\kappa]})$ that can be transformed into $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ with corresponding affine maps $(P_{x,i}, l_{x,i}, P_{u,i}, l_{u,i})$.

Definition 4.4.5 (∞ -Diversity). *The privacy of the actual system state $x_{i,[1,\kappa]}$ and input $u_{i,[1,\kappa]}$ of CAVs is preserved if the cardinality of the set $\Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ is infinite for any feasible state sequence $\bar{x}_{i,[1,\kappa]}$ and input sequence $\bar{u}_{i,[1,\kappa]}$.*

The ∞ -Diversity privacy definition requires that there are infinite sets of $(x_{i,[1,\kappa]}, u_{i,[1,\kappa]})$ and $(P_{x,i}, l_{x,i}, P_{u,i}, l_{u,i})$ that can generate the same $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ received by the attacker. As a result, it is impossible for the attacker to use $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ to infer the actual system state and input information.

Remark 4.4.6. *Definition 4.4.5 of ∞ -Diversity can be viewed as an extension to the l -diversity [161] that has been widely adopted in formal privacy analysis on attribute privacy of tabular. Essentially, l -diversity requires that there are at least l different possible values for the privacy sensitive data attributes, and a greater l indicates greater indistinguishability.*

We next show that the proposed affine transformation mechanism can protect the privacy of CAVs based on Definition 4.4.5.

theorem 4.4.7. *Under the affine masking mechanism described in Section 4.4.4.2, the system state and control input of CAVs are ∞ -Diversity, that is, the attacker cannot infer the actual system state $x_i(k)$ and input $u_i(k)$, $\forall i \in \Omega_C$.*

Proof. Based on Definition 4.4.5, we prove Theorem 4.4.7 by showing that under the affine masking scheme, the cardinality of the set $\Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ is infinite. Specifically, given the sequence $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ accessible to the attacker, for an arbitrary affine map $(P'_{x,i}, l'_{x,i}, P'_{u,i}, l'_{u,i})$ such that $P'_{x,i}$ and $P'_{u,i}$ are invertible, a sequence $(x'_{i,[1,\kappa]}, u'_{i,[1,\kappa]})$ can be uniquely determined based on $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ by using $(P'_{x,i}, l'_{x,i}, P'_{u,i}, l'_{u,i})$ as an inverse mapping. As there exists infinitely many such affine maps $(P'_{x,i}, l'_{x,i}, P'_{u,i}, l'_{u,i})$, there exists infinitely many $(x'_{i,[1,\kappa]}, u'_{i,[1,\kappa]})$ such that via proper affine transformations, the attacker will receive the same accessed information: $(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$, which indicates that the cardinality of the set $\Delta(\bar{x}_{i,[1,\kappa]}, \bar{u}_{i,[1,\kappa]})$ is infinite. $\square$

Remark 4.4.8. *Different from the conventional encryption based techniques [154], [155], the proposed affine masking-based privacy-preserved scheme does not involve complicated encryption and decryption procedure, and thus is light-weight in computation and can be easily implemented in a mixed traffic system. Furthermore, the affine masking mechanism can provide strong privacy protection such that the eavesdropper cannot even approximately estimate the interested information $x_i(k)$ and $u_i(k)$ via the exchanged information between the vehicle system and the central unit.*

4.4.6 Numerical Experiments

In this section, we perform numerical simulations to validate the efficacy of the proposed privacy-preserving *DeeP-LCC*. We consider a mixed traffic system consisting of two CAVs and four HDVs. The CAV and HDV indices are $\Omega_C = \{2, 5\}$ and $\Omega_H = \{1, 3, 4, 6\}$, respectively. The OVM model is used to describe the behavior of HDVs, and a noise with uniform distribution of $\mathbb{U}[-0.3, 0.3]$ is added to the HDVs' model. Both the original *DeeP-LCC* and the proposed privacy-preserving *DeeP-LCC* are used in the numerical simulations.

We follow the similar procedure introduced in [90] to collect the offline data (u^d, ϵ^d, y^d) with the sampling interval chosen as $\Delta t = 0.5$ s. The time horizons for the previous data sequence and future data sequence are chosen as $T_{\text{ini}} = 15$ and $N = 30$, respectively. In the *DeeP-LCC* problem (4.20), the weighting factors are selected as $w_s = 0.5$, $w_v = 1$, and $w_u = 0.1$. The lower and upper bounds of output signal of the CAVs are chose as $y_{\min} = \begin{bmatrix} -15, -30 \end{bmatrix}^\top$, $y_{\max} = \begin{bmatrix} 20, 30 \end{bmatrix}^\top$, while the acceleration limits of the CAVs are set as $a_{\min} = -5$, $a_{\max} = 2$. For the proposed privacy-preserving *DeeP-LCC*, CAVs exploit affine transformation maps (4.21) to mask their actual state and input. The affine transformation maps for CAVs (recall that $\Omega_C = \{2, 5\}$) are chosen as

$$P_{x,2} = \begin{bmatrix} \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) \\ \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \end{bmatrix}, l_{x,2} = \begin{bmatrix} 5 \\ 3 \end{bmatrix},$$

$$P_{u,2} = -1.5, l_{u,2} = 1,$$

and

$$P_{x,5} = \begin{bmatrix} \cos(\frac{8\pi}{9}) & -\sin(\frac{8\pi}{9}) \\ \sin(\frac{8\pi}{9}) & \cos(\frac{8\pi}{9}) \end{bmatrix}, l_{x,5} = \begin{bmatrix} 5 \\ 3 \end{bmatrix},$$

$$P_{u,5} = 1.5, l_{u,5} = -1.$$

Furthermore, based on the aforementioned affine maps and the setup for *DeeP-LCC*, the parameters used to formulate the privacy-preserving *DeeP-LCC* (i.e., problem in 4.26) can be obtained (see Section 4.4.4.2 and 4.4.4.3).

4.4.6.1 Scenario A: Comprehensive Acceleration and Deceleration

A comprehensive acceleration and deceleration scenario is designed to validate the capability of the proposed privacy-preserving *DeeP-LCC* in improving traffic performance. In this scenario, the head vehicle takes acceleration or deceleration at different time periods. Both fuel consumption and velocity errors for the vehicles are considered to quantify traffic performance. More specifically, for the i -th vehicle, the fuel consumption rate f_i (mL/s) is calculated based on an instantaneous model in [162], which is given by

$$f_i = \begin{cases} 0.444 + 0.090R_i v_i + [0.054a_i^2 v_i]_{a_i > 0}, & \text{if } R_i > 0, \\ 0.444, & \text{if } R_i \leq 0, \end{cases}$$

with $R_i = 0.333 + 0.00108v_i^2 + 1.200a_i$. Since the first HDV is not influenced by the CAVs, the total fuel consumption rate f of the mixed traffic system is calculated by summing f_i indexed from 2 to 6, i.e., $f = \sum_{i=2}^6 f_i$. The average absolute velocity error (AAVE) is utilized to quantify velocity errors and is obtained by computing the average of $\frac{|v_i(t) - v_0(t)|}{v_0(t)}$ with respect to the simulation time and the vehicle number. Moreover, the original *DeeP-LCC* scheme and the proposed privacy-preserving *DeeP-LCC* under the Hankel matrix structures are tested in this scenario. The number of columns in Hankel matrices are selected as 1950. A standard output-feedback MPC is also tested in this scenario to facilitate a complete comparison.

The simulation results are shown in Fig. 4.9. Note that the velocity response profiles of MPC matrix based *DeeP-LCC* schemes are quite similar to the ones of Hankel matrix

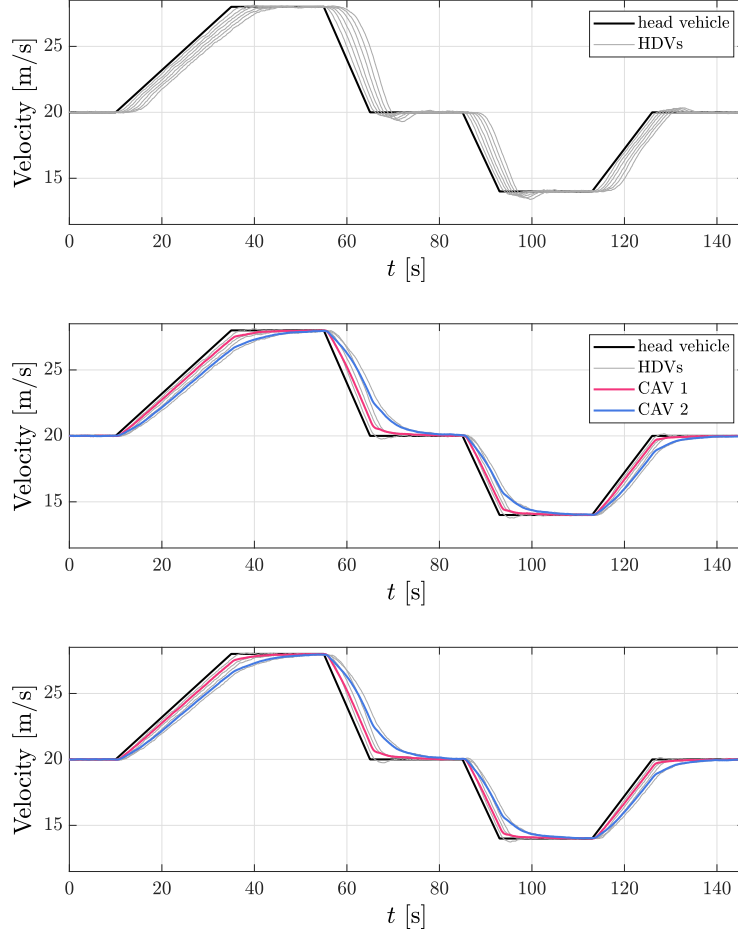


Figure 4.9 Velocity profiles of the mixed traffic system under Scenario A. (a) All the vehicles are HDVs. (b) *DeeP-LCC* under the Hankel matrix structure is utilized to generate the control for CAVs. (c) Privacy-preserving *DeeP-LCC* under the Hankel matrix structure is utilized to generate the control for CAVs.

based *DeeP-LCC* schemes, and hence they are omitted. As shown in Fig. 4.9, compared to the case with all HDVs, both the original *DeeP-LCC* and the proposed privacy-preserving *DeeP-LCC* can effectively mitigate velocity fluctuations, leading to a smoother mixed traffic flow. Table 4.3 presents the fuel consumption and AAVE of different schemes. Note that all *DeeP-LCC* approaches achieve similar performance with MPC, but they exploit pre-collected data to generate control inputs rather than relying on an explicit mixed traffic model. In addition, although affine transformation mechanism is introduced to mask the actual system state and input signals, the proposed privacy-preserving method still retain the advances of

Table 4.3 Fuel Consumption and Average Absolute Velocity Error (AAVE) in Scenario A.

	Fuel Consumption [mL]	AAVE [10^{-3}]
All HDVs	1569.62	27.34
MPC	1536.98(↓2.08%)	24.50(↓10.38%)
DeeP-LCC	1537.98(↓2.02%)	24.52(↓10.29%)
PP-DeeP-LCC	1538.71(↓1.97%)	24.48(↓10.47%)

¹ All the values have been rounded. PP-DeeP-LCC refers to privacy-preserving DeeP-LCC.

original *DeeP-LCC* in improving fuel economy and traffic smoothness.

4.4.6.2 Scenario B: Emergency Braking

We now use a braking scenario to show that the proposed method can protect the privacy of CAVs against the attacker. In this scenario, the head vehicle takes a sudden emergency brake with maximum deceleration, then maintains the low velocity for a while, and finally accelerates to the original velocity.

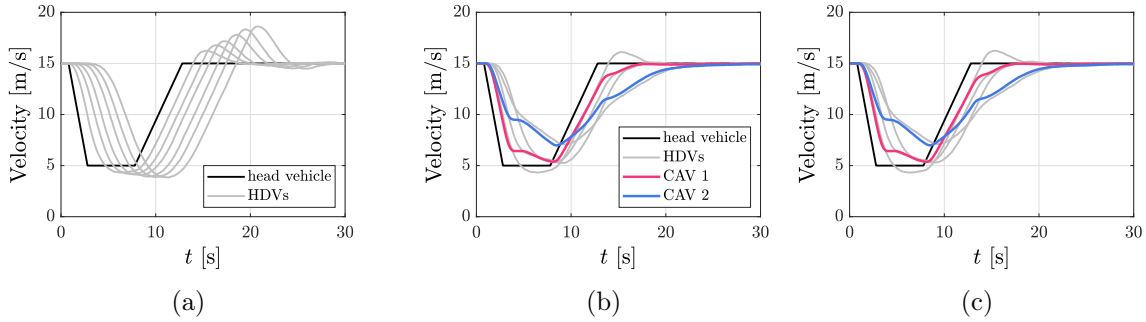


Figure 4.10 Velocity profiles of the mixed traffic system under Scenario B. (a) All the vehicles are HDVs. (b) *DeeP-LCC* under the Hankel matrix structure is utilized to generate the control for CAVs. (c) Privacy-preserving *DeeP-LCC* under the Hankel matrix structure is utilized to generate the control for CAVs.

The simulation results are presented in Figs. 4.10 and 4.11. It can be found from Fig. 4.10 that the CAVs have similar response patterns under the *DeeP-LCC* strategy and the proposed privacy-preserving *DeeP-LCC*. More precisely, when the head vehicle begins to brake (see the time period before 8 s), the CAVs decelerate immediately to adjust the relative distance from the preceding vehicle; when the head vehicle starts to return to the original velocity

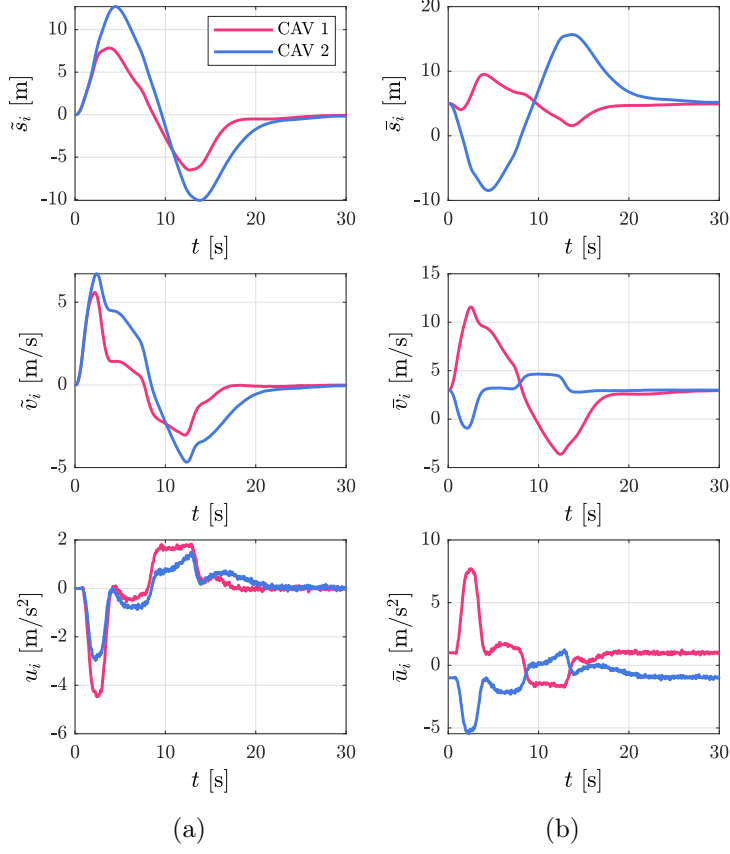


Figure 4.11 Simulation results with privacy-preserving *DeeP-LCC* under Scenario B: (a) actual state and input information of CAVs, (b) state and input information exchanged between the CAVs and the central unit.

(see the time period after 8 s), the CAVs accelerate gradually. The similar motion pattern under these two methods implies that the affine transformation mechanism can retain the control performance as the original *DeeP-LCC* strategy. Meanwhile as shown in Fig. 4.11, under the privacy-preserving *DeeP-LCC*, the exchanged information between the CAVs and the central unit, i.e., $\bar{x}_i(k) = \begin{bmatrix} \bar{s}_i(k), \bar{v}_i(k) \end{bmatrix}^\top$, $\bar{u}_i(k)$, $i \in \Omega_C$, is quite different from the actual one, i.e., $x_i(k) = \begin{bmatrix} \tilde{s}_i(k), \tilde{v}_i(k) \end{bmatrix}^\top$, $u_i(k)$, $i \in \Omega_C$. This indicates that the proposed affine transformation mechanism can conceal the actual state and input of CAVs, which makes the external eavesdropper or the honest-but-curious central unit unable to infer $x_i(k)$ and $u_i(k)$.

4.4.7 Conclusion

This sub-chapter presented a privacy-preserving *Deep-LCC* for CAVs in a mixed-traffic environment. We have considered external eavesdropper and honest-but-curious central unit who intend to infer the CAVs' system states and inputs. A simple yet effective affine transformation mechanism has been designed to enable privacy preservation, and an extended form of the data-enabled predictive control has been derived to achieve safe and optimal control for CAVs. The proposed affine transformation mechanism can be seamlessly integrated into the data-enabled control without affecting the control performance. Theoretical and simulation results confirm that by using the proposed method, the leading cruise control problem in a mixed traffic can be addressed while avoiding disclosing private information to the attacker.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Lucia Lo Bello, Riccardo Mariani, Saad Mubeen, et al. “Recent advances and trends in on-board embedded and networked automotive systems”. In: *IEEE Transactions on Industrial Informatics* 15.2 (2018), pp. 1038–1051.
- [2] Gaetan Kerschen, Keith Worden, Alexander F Vakakis, et al. “Past, present and future of nonlinear system identification in structural dynamics”. In: *Mechanical systems and signal processing* 20.3 (2006), pp. 505–592.
- [3] Jeremy Coulson, John Lygeros, and Florian Dörfler. “Data-Enabled Predictive Control: In the Shallows of the DeePC”. in: *2019 18th European Control Conference (ECC)*. 2019, pp. 307–312.
- [4] Alessandro Chiuso and Gianluigi Pillonetto. “System identification: A machine learning perspective”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 281–304.
- [5] Xia Hong, Richard J Mitchell, Sheng Chen, et al. “Model selection approaches for nonlinear system identification: a review”. In: *International journal of systems science* 39.10 (2008), pp. 925–946.
- [6] Ivan Markovsky, Jan C. Willems, Sabine Van Huffel, et al. *Exact and Approximate Modeling of Linear Systems*. Society for Industrial and Applied Mathematics, 2006.
- [7] Zhaojian Li and Dimitar Filev. “Online Nonlinear System Identification with Evolving Spatial-Temporal Filters”. In: *2018 Annual American Control Conference (ACC)*. 2018, pp. 5274–5279.
- [8] Kaian Chen, Zhaojian Li, Dimitar Filev, et al. “Online Nonlinear Dynamic System Identification With Evolving Spatial-Temporal Filters: Case Study on Turbocharged Engine Modeling”. In: *IEEE Transactions on Control Systems Technology* (2020), pp. 1–8.
- [9] Kaian Chen, Zhaojian Li, Yan Wang, et al. “Online Nonlinear System Identification With Parameter Constraints: Application to Automotive Engine Systems”. In: *Dynamic Systems and Control Conference*. Vol. 59155. American Society of Mechanical Engineers. 2019, V002T11A002.
- [10] Kaian Chen, Kaixiang Zhang, Zhaojian Li, et al. “Stochastic Model Predictive Control for Quasi-Linear Parameter Varying Systems: Case Study on Automotive Engine Control”. In: *Journal of Dynamic Systems, Measurement, and Control* 144.6 (2022), p. 061005.

- [11] Kaian Chen, Kaixiang Zhang, Yusheng Zheng, et al. “Data Enabled Predictive Control for Fast Charging of a Lithium-Ion Battery With Safety Consideration”. In: *Under Preparation* ().
- [12] Kaixiang Zhang, Kaian Chen, Zhaojian Li, et al. “Privacy-Preserving Data-Enabled Predictive Leading Cruise Control in Mixed Traffic”. In: *arXiv preprint arXiv:2205.10916* (2022).
- [13] Nader Raeie, Sajjad Emami, and Omid Karimi Sadaghiyani. “Effects of injection timing, before and after top dead center on the propulsion and power”. In: *Propulsion and Power Research* 3.2 (2014), pp. 59–67. ISSN: 2212-540X.
- [14] Lennart Ljung, ed. *System Identification (2Nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999. ISBN: 0-13-656695-2.
- [15] Jing Sun, I. Kolmanovsky, J. A. Cook, et al. “Modeling and control of automotive powertrain systems: a tutorial”. In: *2015 American Control Conference*. June 2005, pp. 3271–3283.
- [16] X. Yin and J. Liu. “Distributed moving horizon state estimation of two-time-scale nonlinear systems”. In: *Automatica* 79 (2017), pp. 152–161.
- [17] J. W. Grizzle, C. H. Moog, and C. Chevallereau. “Nonlinear control of mechanical systems with an unactuated cyclic variable”. In: *IEEE Transactions on Automatic Control* 50.5 (May 2005), pp. 559–576. ISSN: 0018-9286.
- [18] Vijay M. Janakiraman. “Machine Learning for Identification and Optimal Control of Advanced Automotive Engines”. PhD thesis. The University of Michigan, Ann Arbor, 2013.
- [19] S. A. Serikov. “Neural network model of internal combustion engine”. In: *Cybernetics and Systems Analysis* 46.6 (Nov. 2010), pp. 998–1007.
- [20] Y He and C J Rutland. “Application of artificial neural networks in engine modelling”. In: *International Journal of Engine Research* 5.4 (2004), pp. 281–296.
- [21] L. Brzozowska, K. Brzozowski, and J. Nowakowski. “An Application of Artificial Neural Network to Diesel Engine Modelling”. In: *2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. Sept. 2005, pp. 142–146.
- [22] Christoph Hametner and Stefan Jakubek. “Local model network identification for online engine modelling”. In: *Information Sciences* 220 (2013). Online Fuzzy Machine Learning and Data Mining, pp. 210–225. ISSN: 0020-0255.

- [23] G. Gregorcic and G. Lightbody. “Local Model Network Identification With Gaussian Processes”. In: *IEEE Transactions on Neural Networks* 18.5 (Sept. 2007), pp. 1404–1423.
- [24] Christoph Hametner, Christian Mayr, and Stefan Jakubek. “Dynamic NOx emission modelling using local model networks”. In: *International Journal of Engine Research* 15.8 (2014), pp. 928–933.
- [25] T. Takagi and M. Sugeno. “Fuzzy identification of systems and its applications to modeling and control”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 15.1 (Jan. 1985), pp. 116–132. ISSN: 0018-9472.
- [26] P. P. Angelov and D. P. Filev. “An approach to online identification of Takagi-Sugeno fuzzy models”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.1 (Feb. 2004), pp. 484–498. ISSN: 1083-4419.
- [27] Valentina Breschi, Dario Piga, and Alberto Bemporad. “Piecewise affine regression via recursive multiple least squares and multicategory discrimination”. In: *Automatica* 73 (2016), pp. 155–162. ISSN: 0005-1098.
- [28] L. Bako, K. Boukharouba, E. Duviella, et al. “A recursive identification algorithm for-switched linear/affine models”. In: *Nonlinear Analysis: Hybrid Systems*. 5.3 (2010), pp. 242–253.
- [29] Max A. Woodbury. “Inverting Modified Matrices”. In: Statistical Research Group Memorandum Reports 42. Princeton, NJ: Princeton University, 1950.
- [30] Charles C Holt. “Forecasting seasonals and trends by exponentially weighted moving averages”. In: *International journal of forecasting* 20.1 (2004), pp. 5–10.
- [31] Billy M Williams, Priya K Durvasula, and Donald E Brown. “Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models”. In: *Transportation Research Record* 1644.1 (1998), pp. 132–141.
- [32] Ai Hui Tan and Keith Godfrey. “Introduction: Perturbation Signal Design and Applications”. In: Jan. 2019, pp. 1–24. ISBN: 978-3-030-03660-7.
- [33] D. Q. Mayne. “Model predictive control: Recent developments and future promise”. In: *Automatica* 50 (2014), pp. 2967–2986.
- [34] Roland Tóth. *Modeling and identification of linear parameter-varying systems*. Berlin Springer, 2010.

- [35] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. “Robust constrained model predictive control using linear matrix inequalities”. In: *Automatica* (1996), pp. 1361–1379.
- [36] Tae Hyoung Kim, Jee Hun Park, and Toshiharu Sugie. “Output-feedback model predictive control for LPV systems with input saturation based on quasi-min-max algorithm”. In: *Proceedings of the IEEE Conference on Decision and Control*. 2006, pp. 1454–1459.
- [37] Hossam S. Abbas, Roland Tóth, Nader Meskin, et al. “A Robust MPC for Input-Output LPV Models”. In: *IEEE Transactions on Automatic Control* 61 (2016), pp. 4183–4188.
- [38] Alessandro Casavola and Domenico Famularo. “A Feedback Min-Max MPC Algorithm for LPV Systems Subject to Bounded Rates of Change of Parameters”. In: *IEEE Transactions on Automatic Control* 47 (2002), pp. 1147–1153.
- [39] Dewei Li and Yugeng Xi. “The Feedback Robust MPC for LPV Systems With Bounded Rates of Parameter Changes”. In: *IEEE Transactions on Automatic Control* 55 (2010), pp. 503–507.
- [40] Hossam Seddik Abbas, Georg Männel, Christian Herzog né Hoffmann, et al. “Tube-based model predictive control for linear parameter-varying systems with bounded rate of parameter variation”. In: *Automatica* (2019), pp. 21–28.
- [41] Ali Mesbah. “Stochastic model predictive control: An overview and perspectives for future research”. In: *IEEE Control Systems* 36 (2016), pp. 30–44.
- [42] Marcello Farina, Luca Giulioni, and Riccardo Scattolini. “Stochastic linear Model Predictive Control with chance constraints - A review”. In: *Journal of Process Control* 44 (2016), pp. 53–67.
- [43] Mark Cannon, Basil Kouvaritakis, and Xingjian Wu. “Probabilistic Constrained MPC for Multiplicative and Additive Stochastic Uncertainty”. In: *IEEE Transactions on Automatic Control* 54 (2009), pp. 1626–1632.
- [44] Mark Cannon, Basil Kouvaritakis, and Desmond Ng. “Probabilistic tubes in linear stochastic model predictive control”. In: *Systems and Control Letters* 58 (2009), pp. 747–753.
- [45] Mark Cannon, Basil Kouvaritakis, Saša V Raković, et al. “Stochastic tubes in model predictive control with probabilistic constraints”. In: *IEEE Transactions on Automatic Control* 56 (2011), pp. 194–200.

- [46] Basil Kouvaritakis, Mark Cannon, Saša V Raković, et al. “Explicit use of probabilistic distributions in linear predictive control”. In: *Automatica* 46 (2010), pp. 1719–1724.
- [47] Daniele Bernardini and Alberto Bemporad. “Scenario-based model predictive control of stochastic constrained linear systems”. In: *48th IEEE Conference on Decision and Control*. 2009.
- [48] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, et al. “The scenario approach for Stochastic Model Predictive Control with bounds on closed-loop constraint violations”. In: *Automatica* 50 (2014), pp. 3009–3018.
- [49] Lukas Hewing and Melanie N. Zeilinger. “Scenario-Based Probabilistic Reachable Sets for Recursively Feasible Stochastic Model Predictive Control”. In: *IEEE Control Systems Letters* (2020), pp. 450–455.
- [50] Shaikshavali Chitraganti, Roland Toth, Nader Meskin, et al. “Stochastic model predictive control for LPV systems”. In: *Proceedings of the American Control Conference*. 2017, pp. 5654–5659.
- [51] Giuseppe C. Calafiore and Lorenzo Fagiano. “Stochastic model predictive control of LPV systems via scenario optimization”. In: *Automatica* 49 (2013), pp. 1861–1866.
- [52] Pablo S.G. Cisneros, Sophia Voss, and Herbert Werner. “Efficient Nonlinear Model Predictive Control via quasi-LPV representation”. In: *2016 IEEE 55th Conference on Decision and Control*. 2016, pp. 3216–3221.
- [53] Pablo Gonzalez Cisneros and Herbert Werner. “Fast Nonlinear MPC for Reference Tracking Subject to Nonlinear Constraints via Quasi-LPV Representations”. In: *IFAC-PapersOnLine* (2017), pp. 11601–11606.
- [54] Marcelo Menezes Morato, Julio E. Normey-Rico, and Olivier Sename. “Novel qLPV MPC Design with Least-Squares Scheduling Prediction”. In: *IFAC-PapersOnLine* 52 (2019), pp. 158–163.
- [55] Naresh N. Nandola and Sharad Bhartiya. “A multiple model approach for predictive control of nonlinear hybrid systems”. In: *Journal of Process Control* 18 (2008), pp. 131–148.
- [56] Pablo S.G. Cisneros, Aadithyan Sridharan, and Herbert Werner. “Constrained Predictive Control of a Robotic Manipulator using quasi-LPV Representations”. In: *IFAC-PapersOnLine* 51 (2018), pp. 118–123.
- [57] Marcelo Menezes Morato, Julio Elias Normey-Rico, and Olivier Sename. “Sub-optimal recursively feasible linear parameter-varying predictive algorithm for semi-active suspension control”. In: *IET Control Theory and Applications* 14 (2020),

pp. 2764–2775.

- [58] Lukas Hewing and Melanie N. Zeilinger. “Stochastic Model Predictive Control for Linear Systems Using Probabilistic Reachable Sets”. In: *Proceedings of the IEEE Conference on Decision and Control*. 2018, pp. 5182–5188.
- [59] David Q. Mayne and W. Langson. “Robustifying model predictive control of constrained linear systems”. In: *Electronics Letters* 37 (2001), pp. 1422–1423.
- [60] Goncalo Collares Pereira, Pedro F. Lima, Bo Wahlberg, et al. “Linear Time-Varying Robust Model Predictive Control for Discrete-Time Nonlinear Systems”. In: *Proceedings of the IEEE Conference on Decision and Control*. 2018, pp. 2659–2666.
- [61] J. C. Geromel, P. L.D. Peres, and J. Bernussou. “On a convex parameter space method for linear control design of uncertain systems”. In: *SIAM Journal on Control and Optimization* 29 (1991), pp. 381–402.
- [62] Lukas Hewing, Kim P. Wabersich, and Melanie N. Zeilinger. “Recursively feasible stochastic model predictive control using indirect feedback”. In: *Automatica* 119 (2020).
- [63] Xinjia Chen. “A New Generalization of Chebyshev Inequality for Random Vectors”. In: *arXiv preprint arXiv:0707.0805* (2007).
- [64] Alberto Bemporad, Daniele Bernardini, Ruixing Long, et al. “Model Predictive Control of Turbocharged Gasoline Engines for Mass Production”. In: *SAE Technical Papers* (2018), pp. 1–12.
- [65] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl, eds. *Model Predictive Control: Theory, Computation, and Design - 2nd Edition*. Madison: Nob Hill Publishing, 2017.
- [66] M. Garza-Fabre, G. T. Pulido, and C. A. C. Coello. “Ranking methods for many-objective optimization”. In: *MICAI 2009: Advances in Artificial Intelligence* (2009), pp. 633–645.
- [67] Pornchai Bumroongsri. “Tube-based robust MPC for linear time-varying systems with bounded disturbances”. In: *International Journal of Control, Automation and Systems* 13 (2015), pp. 620–625.
- [68] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, et al. “Invariant approximation of the minimal robust positively invariant set”. In: *IEEE Transactions on Automatic Control* 50.3 (2005), pp. 406–410.

- [69] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, et al. “A note on persistency of excitation”. In: *Systems & Control Letters* 54.4 (2005), pp. 325–329. ISSN: 0167-6911.
- [70] Felix Fiedler and Sergio Lucia. “On the relationship between data-enabled predictive control and subspace predictive control”. In: *2021 European Control Conference (ECC)*. 2021, pp. 222–229.
- [71] Jeremy Coulson, John Lygeros, and Florian Dörfler. “Regularized and Distributionally Robust Data-Enabled Predictive Control”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019, pp. 2696–2701.
- [72] Florian Dorfler, Jeremy Coulson, and Ivan Markovsky. “Bridging direct and indirect data-driven control formulations via regularizations and relaxations”. In: *IEEE Transactions on Automatic Control* (2022), pp. 1–1.
- [73] Alberto Sanfeliu, Maria Rosa Llàcer, Maria Dolors Gramunt, et al. “Influence of the privacy issue in the deployment and design of networking robots in European urban areas”. In: *Advanced Robotics* 24.13 (2010), pp. 1873–1899.
- [74] Jonathan P West, Casey A Klostad, Joseph E Uscinski, et al. “Citizen support for domestic drone use and regulation”. In: *American Politics Research* 47.1 (2019), pp. 119–151.
- [75] J. Cortés, G. E. Dullerud, S. Han, et al. “Differential privacy in control and network systems”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. Dec. 2016, pp. 4252–4272.
- [76] Oded Goldreich. “Foundation of cryptography (in two volumes: Basic tools and basic applications)”. In: (2001).
- [77] R. A. Popa, H. Balakrishnan, and A. J. Blumberg. “VPriv: protecting Privacy in Location-Based Vehicular Services”. In: *Proceedings of USENIX Security Symposium*. 2009, pp. 335–350.
- [78] B. Hoh, T. Iwuchukwu, Q. Jacobson, et al. “Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines”. In: *IEEE Transactions on Mobile Computing* 11.5 (2012), pp. 849–864.
- [79] B. Hoh, M. Gruteser, H. Xiong, et al. “Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking”. In: *IEEE Transactions on Mobile Computing* 9.8 (2010), pp. 1089–1107.
- [80] S. Gisdakis, V. Manolopoulos, S. Tao, et al. “Secure and privacy-preserving smartphone-based traffic information systems”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.3 (2015), pp. 1428–1438.

- [81] J. Zhang and C. Y. Chow. “REAL: a reciprocal protocol for location privacy in wireless sensor networks”. In: *IEEE Transactions on Dependable and Secure Computing* 12.4 (2015), pp. 458–471.
- [82] S. U. Hussain and F. Koushanfar. “Privacy preserving localization for smart automotive systems”. In: *Proceedings of the 53rd Annual Design Automation Conference*. 2016, pp. 26–31.
- [83] P. Kalnis, G. Ghinita, K. Mouratidis, et al. “Preventing location-based identity inference in anonymous spatial queries”. In: *IEEE Transactions on Knowledge and Data Engineering* 19.12 (2007), pp. 1719–1733.
- [84] M. Ghaffari, N. Ghadiri, M. H. Manshaei, et al. “P⁴QS: A Peer to Peer Privacy Preserving Query Service for Location-Based Mobile Applications”. In: *IEEE Transactions on Vehicular Technology* in press (2017).
- [85] Z. Yang, S. Yu, W. Lou, et al. “P²: Privacy-preserving communication and precise reward architecture for V2G networks in smart grid”. In: *IEEE Transactions on Smart Grid* 2.4 (2011), pp. 697–706.
- [86] L. P. Clare, J. L. Gao, E. H. Jennings, et al. “A network architecture for precision formation flying using the IEEE 802.11 MAC protocol”. In: *Proceedings of 2005 IEEE Aerospace conference*. 2005, pp. 1335–1347.
- [87] NASA Research Announcement. “New Millennium Program Space Technology-9 (ST-9)”. In: (2004).
- [88] Paolo Gherardo Carlet, Andrea Favato, Saverio Bolognani, et al. “Data-Driven Continuous-Set Predictive Current Control for Synchronous Motor Drives”. In: *IEEE Transactions on Power Electronics* 37.6 (2022), pp. 6637–6646.
- [89] Linbin Huang, Jeremy Coulson, John Lygeros, et al. “Decentralized Data-Enabled Predictive Control for Power System Oscillation Damping”. In: *IEEE Transactions on Control Systems Technology* (2021), pp. 1–13.
- [90] Jiawei Wang, Yang Zheng, Qing Xu, et al. *Data-Driven Predictive Control for Connected and Autonomous Vehicles in Mixed Traffic*. 2021.
- [91] Jeremy Coulson, John Lygeros, and Florian Dorfler. “Distributionally robust chance constrained data-enabled predictive control”. In: *IEEE Transactions on Automatic Control* (2021).
- [92] Ezzat Elokda, Jeremy Coulson, Paul N Beuchat, et al. “Data-enabled predictive control for quadcopters”. In: *International Journal of Robust and Nonlinear Control* 31.18 (2021), pp. 8916–8936.

- [93] Michel Armand and J-M Tarascon. “Building better batteries”. In: *nature* 451.7179 (2008), pp. 652–657.
- [94] Sheng S Zhang, Kang Xu, and TR Jow. “Study of the charging process of a LiCoO₂-based Li-ion battery”. In: *Journal of power sources* 160.2 (2006), pp. 1349–1354.
- [95] Sheng Shui Zhang. “The effect of the charging protocol on the cycle life of a Li-ion battery”. In: *Journal of power sources* 161.2 (2006), pp. 1385–1391.
- [96] Fuqiang An, Rui Zhang, Zhiguo Wei, et al. “Multi-stage constant-current charging protocol for a high-energy-density pouch cell based on a 622NCM/graphite system”. In: *RSC advances* 9.37 (2019), pp. 21498–21506.
- [97] Asghar Aryanfar, Daniel Brooks, Boris V Merinov, et al. “Dynamics of lithium dendrite growth and inhibition: Pulse charging experiments and Monte Carlo calculations”. In: *The journal of physical chemistry letters* 5.10 (2014), pp. 1721–1726.
- [98] Jun Li, Edward Murphy, Jack Winnick, et al. “The effects of pulse charging on cycling characteristics of commercial lithium-ion batteries”. In: *Journal of Power Sources* 102.1-2 (2001), pp. 302–309.
- [99] Thanh Tu Vo, Xiaopeng Chen, Weixiang Shen, et al. “New charging strategy for lithium-ion batteries based on the integration of Taguchi method and state of charge estimation”. In: *Journal of Power Sources* 273 (2015), pp. 413–422.
- [100] Habiballah Rahimi-Eichi, Federico Baronti, and Mo-Yuen Chow. “Online adaptive parameter identification and state-of-charge coestimation for lithium-polymer battery cells”. In: *IEEE Transactions on Industrial Electronics* 61.4 (2013), pp. 2053–2061.
- [101] Robin Drees, Frank Lienesch, and Michael Kurrat. “Fast charging lithium-ion battery formation based on simulations with an electrode equivalent circuit model”. In: *Journal of Energy Storage* 36 (2021), p. 102345.
- [102] Caiping Zhang, Jiuchun Jiang, Yang Gao, et al. “Charging optimization in lithium-ion batteries based on temperature rise and charge time”. In: *Applied energy* 194 (2017), pp. 569–577.
- [103] Zheng Chen, Bing Xia, Chunting Chris Mi, et al. “Loss-minimization-based charging strategy for lithium-ion battery”. In: *IEEE Transactions on Industry Applications* 51.5 (2015), pp. 4121–4129.
- [104] Zhen Guo, Bor Yann Liaw, Xinping Qiu, et al. “Optimal charging method for lithium ion batteries using a universal voltage protocol accommodating aging”. In: *Journal of Power Sources* 274 (2015), pp. 957–964.

- [105] Shun-Chung Wang and Yi-Hua Liu. “A PSO-based fuzzy-controlled searching for the optimal charge pattern of Li-ion batteries”. In: *IEEE Transactions on Industrial Electronics* 62.5 (2014), pp. 2983–2993.
- [106] Yasha Parvini and Ardalan Vahidi. “Maximizing charging efficiency of lithium-ion and lead-acid batteries using optimal control theory”. In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 317–322.
- [107] Marc Doyle, Thomas F Fuller, and John Newman. “Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell”. In: *Journal of the Electrochemical society* 140.6 (1993), p. 1526.
- [108] Nalin A Chaturvedi, Reinhardt Klein, Jake Christensen, et al. “Algorithms for advanced battery-management systems”. In: *IEEE Control systems magazine* 30.3 (2010), pp. 49–68.
- [109] Reinhardt Klein, Nalin A Chaturvedi, Jake Christensen, et al. “Optimal charging strategies in lithium-ion battery”. In: *Proceedings of the 2011 american Control Conference*. IEEE. 2011, pp. 382–387.
- [110] Resmi Suresh and Raghunathan Rengaswamy. “Modeling and control of battery systems. Part II: A model predictive controller for optimal charging”. In: *Computers & Chemical Engineering* 119 (2018), pp. 326–335.
- [111] Yilin Yin, Yang Hu, Song-Yul Choe, et al. “New fast charging method of lithium-ion batteries based on a reduced order electrochemical model considering side reaction”. In: *Journal of Power Sources* 423 (2019), pp. 367–379.
- [112] Yilin Yin and Song-Yul Choe. “Actively temperature controlled health-aware fast charging method for lithium-ion battery using nonlinear model predictive control”. In: *Applied Energy* 271 (2020), p. 115232.
- [113] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. “Singular value decomposition and principal component analysis”. In: *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.
- [114] Changfu Zou, Xiaosong Hu, Zhongbao Wei, et al. “Electrochemical estimation and control for lithium-ion battery health-aware fast charging”. In: *IEEE Transactions on Industrial Electronics* 65.8 (2017), pp. 6635–6645.
- [115] Tobias C Bach, Simon F Schuster, Elena Fleder, et al. “Nonlinear aging of cylindrical lithium-ion cells linked to heterogeneous compression”. In: *Journal of Energy Storage* 5 (2016), pp. 212–223.

- [116] Anna Tomaszewska, Zhengyu Chu, Xuning Feng, et al. “Lithium-ion battery fast charging: A review”. In: *ETransportation* 1 (2019), p. 100011.
- [117] Marcello Torchio, Nicolas A Wolff, Davide M Raimondo, et al. “Real-time model predictive control for the optimal charging of a lithium-ion battery”. In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 4536–4541.
- [118] Yufang Lu, Xuebing Han, Guangjin Zhao, et al. “Optimal Charging of Lithium-ion Batteries Based on Model Predictive Control Considering Lithium Plating and Cell Temperature”. In: *2021 6th International Conference on Power and Renewable Energy (ICPRE)*. IEEE. 2021, pp. 1248–1253.
- [119] Henk J van Waarde, Claudio De Persis, M Kanat Camlibel, et al. “Willems’ fundamental lemma for state-space systems and its extension to multiple datasets”. In: *IEEE Control Systems Letters* 4.3 (2020), pp. 602–607.
- [120] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.
- [121] Matan Gavish and David L Donoho. “The optimal hard threshold for singular values is $4/\sqrt{3}$ ”. In: *IEEE Transactions on Information Theory* 60.8 (2014), pp. 5040–5053.
- [122] Marcello Torchio, Lalo Magni, R Bhushan Gopaluni, et al. “Lionsimba: a matlab framework based on a finite volume model suitable for li-ion battery design, simulation, and control”. In: *Journal of The Electrochemical Society* 163.7 (2016), A1192.
- [123] Ardalan Vahidi and Antonio Sciarretta. “Energy saving potentials of connected and automated vehicles”. In: *Transp. Res. Part C: Emerg. Technol.* 95 (2018), pp. 822–843.
- [124] Shengbo Eben Li, Yang Zheng, Keqiang Li, et al. “Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities”. In: *IEEE Intell. Transp. Syst. Mag.* 9.3 (2017), pp. 46–58.
- [125] Bart Van Arem, Cornelie J. G. Van Driel, and Ruben Visser. “The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics”. In: 7.4 (2006), pp. 429–436.
- [126] Vicente Milanés, Steven E Shladover, John Spring, et al. “Cooperative adaptive cruise control in real traffic situations”. In: 15.1 (2013), pp. 296–305.
- [127] Yiheng Feng, K Larry Head, Shayan Khoshmaghham, et al. “A real-time adaptive signal control in a connected vehicle environment”. In: *Transp. Res. Part C: Emerg. Technol.* 55 (2015), pp. 460–473.

- [128] Yang Zheng, Shengbo Eben Li, Keqiang Li, et al. “Distributed Model Predictive Control for Heterogeneous Vehicle Platoons Under Unidirectional Topologies”. In: 25.3 (2017), pp. 899–910.
- [129] Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, et al. “Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments”. In: *Transp. Res. Part C: Emerg. Technol.* 89 (2018), pp. 205–221.
- [130] Chaojie Wang, Siyuan Gong, Anye Zhou, et al. “Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints”. In: *Transp. Res. Part C: Emerg. Technol.* 113 (2020), pp. 124–145.
- [131] Hongyan Guo, Jun Liu, Qikun Dai, et al. “A distributed adaptive triple-step nonlinear control for a connected automated vehicle platoon with dynamic uncertainty”. In: *IEEE Internet Things J.* 7.5 (2020), pp. 3861–3871.
- [132] Gábor Orosz. “Connected cruise control: modelling, delay effects, and nonlinear behaviour”. In: *Veh. Syst. Dyn.* 54.8 (2016), pp. 1147–1176.
- [133] I Ge Jin and Gábor Orosz. “Optimal control of connected vehicle systems with communication delay and driver reaction time”. In: 18.8 (2017), pp. 2056–2070.
- [134] I Ge Jin, Sergei S Avedisov, Chaozhe R He, et al. “Experimental validation of connected automated vehicle design among human-driven vehicles”. In: *Transp. Res. Part C: Emerg. Technol.* 91 (2018), pp. 335–352.
- [135] Jiawei Wang, Yang Zheng, Chaoyi Chen, et al. “Leading cruise control in mixed traffic flow: System modeling, controllability, and string stability”. In: (in press, 2021).
- [136] Shuo Feng, Ziyu Song, Zhaojian Li, et al. “Robust platoon control in mixed traffic flow based on tube model predictive control”. In: 6.4 (2021), pp. 711–722.
- [137] Yang Zheng, Jiawei Wang, and Keqiang Li. “Smoothing traffic flow via control of autonomous vehicles”. In: *IEEE Internet Things J.* 7.5 (2020), pp. 3882–3896.
- [138] Jiawei Wang, Yang Zheng, Qing Xu, et al. “Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles”. In: 22.12 (2020), pp. 7445–7459.
- [139] Masako Bando, Katsuya Hasebe, Akihiro Nakayama, et al. “Dynamical model of traffic congestion and numerical simulation”. In: *Phys. Rev. E* 51.2 (1995), p. 1035.
- [140] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Phys. Rev. E* 62.2 (2000), p. 1805.

- [141] Tianshu Chu and Uroš Kalabić. “Model-based deep reinforcement learning for CACC in mixed-autonomy vehicle platoon”. In: 2019, pp. 4079–4084.
- [142] Eugene Vinitsky, Kanaad Parvate, Aboudy Kreidieh, et al. “Lagrangian control through Deep-RL: Applications to bottleneck decongestion”. In: 2018, pp. 759–765.
- [143] Dong Chen, Zhaojian Li, Yongqiang Wang, et al. “Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic”. In: *arXiv preprint arXiv:2105.05701* (2021).
- [144] Weinan Gao, Zhong-Ping Jiang, and Kaan Ozbay. “Data-driven adaptive optimal control of connected vehicles”. In: 18.5 (2016), pp. 1122–1133.
- [145] Mengzhe Huang, Zhong-Ping Jiang, and Kaan Ozbay. “Learning-based adaptive optimal control for connected vehicles in mixed traffic: robustness to driver reaction time”. In: (in press, 2020).
- [146] Tong Liu, Leilei Cui, Bo Pang, et al. “Data-Driven Adaptive Optimal Control of Mixed-Traffic Connected Vehicles in a Ring Road”. In: 2021, pp. 77–82.
- [147] Lukas Hewing, Kim P Wabersich, Marcel Menner, et al. “Learning-based model predictive control: Toward safe learning in control”. In: *Annu. Rev. Control Robot. Auton. Syst.* 3 (2020), pp. 269–296.
- [148] Julian Berberich, Johannes Köhler, Matthias A Müller, et al. “Data-driven model predictive control with stability and robustness guarantees”. In: 66.4 (2021), pp. 1702–1717.
- [149] Jiawei Wang, Yang Zheng, Jianghong Dong, et al. “Experimental Validation of DeeP-LCC for Dissipating Stop-and-Go Waves in Mixed Traffic”. In: *arXiv preprint arXiv:2204.03747* (2022).
- [150] Ivan Markovsky and Florian Dörfler. “Behavioral systems theory in data-driven analysis, signal processing, and control”. In: *Annu. Rev. Control* 52 (2021), pp. 42–64.
- [151] Jonathan Petit and Steven E. Shladover. “Potential Cyberattacks on Automated Vehicles”. In: 16.2 (2015), pp. 546–556.
- [152] Mani Amoozadeh, Arun Raghuramu, Chen-Nee Chuah, et al. “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving”. In: 53.6 (2015), pp. 126–132.
- [153] Faezeh Farivar, Mohammad Sayad Haghighi, Alireza Jolfaei, et al. “On the security of networked control systems in smart vehicle and its adaptive cruise control”. In:

- 22.6 (2021), pp. 3824–3831.
- [154] Moritz Schulze Darup, Adrian Redder, Iman Shames, et al. “Towards encrypted MPC for linear constrained systems”. In: 2.2 (2017), pp. 195–200.
 - [155] Andreea B Alexandru, Manfred Morari, and George J Pappas. “Cloud-based MPC with encrypted data”. In: 2018, pp. 5014–5019.
 - [156] Kaixiang Zhang, Kaian Chen, Zhaojian Li, et al. “Privacy-Preserved Data-Enabled Predictive Control for Connected and Automated Vehicles in Mixed Traffic”. In: under review, 2022.
 - [157] A. A. H. Damen, P. M. J. Van den Hof, and A. K. Hajdasinski. “Approximate realization based upon an alternative to the Hankel matrix: the Page matrix”. In: 2.4 (1982), pp. 202–208.
 - [158] Alimzhan Sultangazin and Paulo Tabuada. “Symmetries and isomorphisms for privacy in control over the cloud”. In: 66.2 (2020), pp. 538–549.
 - [159] Kaixiang Zhang, Zhaojian Li, Yongqiang Wang, et al. “Privacy-Preserved Nonlinear Cloud-based Model Predictive Control via Affine Masking”. In: *arXiv preprint arXiv:2112.10625* (2021).
 - [160] Huan Gao, Zhaojian Li, and Yongqiang Wang. “Privacy-Preserving Collaborative Estimation for Networked Vehicles With Application to Collaborative Road Profile Estimation”. In: (in press, 2022). DOI: 10.1109/TITS.2022.3154650.
 - [161] A. Machanavajjhala, D. Kifer, J. Gehrke, et al. “l-diversity: Privacy beyond k-anonymity”. In: *ACM Trans. Knowl. Discov. Data* 1.1 (2007), pp. 3–14.
 - [162] Darrell P. Bowyer, Rahmi Akçelik, and D. C. Biggs. *Guide to fuel consumption analyses for urban traffic management*. 1985.