COVERAGE CONTROL FOR HETEROGENEOUS MULTI-AGENT SYSTEMS

by

FARSHID ABBASI

(Under the Direction of Javad Mohammadpour)

ABSTRACT

In the present dissertation, a team-based coverage control is proposed that aims at deploying groups of agents to an environment with a probability distribution function representing the likelihood of an event in different regions. The proposed approach can handle the deployment of heterogeneous teams agents each of which pursing a different objective or assigned task. The presented approaches are then implemented on a set of numerical examples to asses their performance. As the next task and in a different domain, the development of robust nonlinear control techniques is studied for uncertain systems. Various types of uncertainty is investigated in three major areas; first, a robust identification approach is proposed for the *Linear Parameter Varying* (LPV) identification of nonlinear systems with uncertain Scheduling variables. Two deterministic and stochastic techniques are developed and their performance are compared with the previous methods in the literature. Next, a robust reduced-order model based controller is designed for a system represented by a parabolic *Partial Differential Equation* (PDE). The objective is to take into account the variation of the model parameter and its effect on the reduced-order model. Then, the reduced model is used to design a robust nonlinear controller to control the main full-order model.

INDEX WORDS: Multi-agent Systems, heterogeneous agents, coverage control, team-based coverage, Linear Parameter-varying Models, System Identification, Robust Control, Nonlinear Control

COVERAGE CONTROL FOR HETEROGENEOUS MULTI-AGENT SYSTEMS

by

FARSHID ABBASI

B.E., University of Tabriz, Iran, 2007

M. S., University of Tabriz, Iran, 2010

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2016

COVERAGE CONTROL FOR HETEROGENEOUS MULTI-AGENT SYSTEMS

by

FARSHID ABBASI

Approved:

Major Professor: Javad Mohammadpour

Committee: Mark A. Haidekker
Caner Kazanci
Beshah Ayalew

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2016

DEDICATION

To my mother, Parvaneh. Her support, encouragement, and constant love have sustained me throughout my life.

ACKNOWLEDGMENTS

I would like to convey my gratitude and respect for my thesis and research advisor, Professor Javad Mohammadpour Velni. He has inspired me to become an independent researcher and helped me realize the power of critical reasoning. My sincere thanks must also go to the members of my thesis advisory and exam committee: Professors Mark A. Haidekker, Caner Kazanci, and Beshah Ayalew. They generously gave their time to offer me valuable comments toward improving my work. I am most grateful to the collaborators for lending me their expertise and intuition to my scientific and technical problems: Dr. Roland Tóth and Dr. Nader Meskin.

I cannot forget a friend who went through hard times with me, cheered me on, and celebrated each accomplishment. I will certainly remember the long coffee hours and never ending rants from start to the end of this laborious path. I was blessed to get to know great friend of mine Dr. Zeeshan Rizvi and his lovely family, specially, my best buddy Sarah.

My special thanks goes to Sama for all her support. I am blessed to have her by my side throughout my PhD. Understanding me best as a Ph.D. herself, she has been my great companion, supported, encouraged, and helped me get through this agonizing period in the most positive way.

I would like to express my gratitude to my parents for all their support and endless kindness. Their heart-warming kindness has been a great gift and inspiration to reach my academic milestones

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND LITERATURE SURVEY

1.1 LITERATURE SURVEY ON TEAM-BASED COVERAGE CONTROL OF HETEROGENEOUS MOVING SENSOR NETWORKS

Due to the advantages of the distributed systems such as reliability, speed and economics over the centralized systems, the distributed deployment algorithms have been proposed for the workload sharing and partitioning tasks. To this aim, each robot needs to only exchange information collected by its sensors, e.g., position and velocity, with other agents and negotiate its scheduled task with a number of other agents. Then, each robot locally generates an appropriate control action using information gathered from its neighbors. For mobile sensing networks, a distributed control strategy has been proposed in [32, 124] to equally divide an assigned area into subregions, where each robot is able to obtain locations of its neighbors through, e.g., adjust-communication-radius algorithm. Then, it computes the associated Voronoi cell and moves toward the centroid of its Voronoi cell obtained based on the gradient descent method that gives the optimal solution for equal partitioning problem [1, 37, 126].

The existing approaches for the coverage control are based on the assumption that all agents belong to a single team [109]. However, this assumption is not realistic in many real-world applications, as the agents may differ from , e.g., dynamics or communication perspective [135]. A multi-robot system can generally be considered as a homogeneous or heterogeneous system depending on the similarities or differences in their properties, e.g., desired performance index, dynamics, etc., that is required when coping with various complex assigned tasks [65, 134, 155]. In the present work, a new coverage strategy is proposed that aims at taking into account the differences in the robots dynamics by offering a team-based design approach, where, each robot might team up with

others based on its assigned task, associated dynamics or embedded communication capabilities. This would make it possible to improve reliability and flexibility of the deployment algorithm. It should be noted that throughout this work, it is assumed that the structure of the teams and the agents within each team is known a priori.

The present work introduces a new team-based coverage control scheme which can handle different scenarios in heterogeneous systems of robots. The presented approach addresses the problem of the agents deployment by considering teams of robots instead of evaluating each agent individually. The main problem can be defined as a two-level optimization problem, where one problem is defined inside each team and another is defined among the teams in the overall coverage space. The agents will move towards the local minima until the optimum configuration is achieved. In the proposed approach, firstly, a local minimum to the deployment problem is obtained in the team level. Then, a second optimization problem is solved to guarantee the convergence of the agents to their optimum location generating their respective Voronoi cells inside the teams. By considering the nucleus as the associated Voronoi centroid of each team, the optimization problem at the team level is defined to maximize the performance of the team.

1.2 Introduction to Robust and Stochastic Methods for Systems Subjected to Uncertainty

Model identification or control of systems with uncertain or missing inputs or parameters has attracted the attention of researchers due to their critical applications when dealing with real systems where measurements and parameter values are prone to noise and error. The inherent level of difficulty of robust or stochastic techniques obviously depends on the complexity of the system under study. One can distinguish between the necessity of implementation of either stochastic or robust methods in different situations, depending on the nature of a particular application. In the present proposal, we study uncertain systems in three different categories.

First, the model identification of LPV systems is studied when the measured scheduling variables are uncertain. Identification schemes need to overcome the distortions in the data due to

the presence of uncertainty and random stochastic noise. Most of the existing methods for model identification of *linear parameter-varying* (LPV) systems consider the scheduling variables to be noise free. However, the presence of uncertainty, i.e., noise, in the measured data including the scheduling variables is inevitable and can lead to an inaccurate model identification. Hence, the precise knowledge of scheduling variables in the presence of uncertainties is a critical issue in both LPV model identification and LPV control design.

Second, the problem of overcoming the uncertainties is tackled from the control perspective. The control theory is concerned with influencing systems to realize that certain output quantities take a desired course in the presence of measurement and parameter uncertainties. These can be technical systems, like heating a room with output temperature, a boat with the output quantities heading and speed, or a power plant with the output electrical power. The underlying dynamical behaviours can be described by differential equations, difference equations or other functional equations. Among wide variety of the systems governed by partial differential equations, the fluid systems are of great importance due to their many applications like chemical processes, drug delivery and many other. In this work, design of a robust sliding mode controller is studied for fluid systems governed by Burgers' equation with Neumann boundary conditions in the presence of model uncertainties. The main objective is to design a reduced-order model based controller at a nominal value of the system parameter that stabilizes the full order model without the need for online computation or update of the reduced-order model. Two types of model uncertainties are resulted from the variation of the parameter ν; first, the error arising from the change in the state of the full-order model, second, the error associated with the estimated proper orthogonal decomposition (POD) basis functions at the nominal value of ν_0.

Finally, a control problem of a distributed system of agents is studied under presence of uncertainty. The uncertainty represents the probability of desired event happening in the area of interest that needs to be covered and monitored. This is to say agents should take the uncertainty into account while performing their assigned tasks. To this end, uncertainty is taken into account as an incorporated Probability Density Function (PDF) of an event happening in the region in the associ-

ated optimization problem. In this work, we study a new team-based approach when dealing with multiple regions of interest with various probability distribution functions. Therefore, agents divide the main region among themselves to achieve the optimal coverage from the team perspective to deal with multiple levels of uncertainty in the region.

In the following section, we take a look at the existing state of the art in the LPV identification and control of uncertain systems literature pertaining to developing robust and stochastic methods for systems subjected to uncertainty.

1.3 Literature survey on LPV system identification with Uncertain Scheduling Parameters

Accurate knowledge of scheduling signals is a critical assumption in both LPV system identification and LPV control design. The previous works [75, 146] that use the kernel-based SVM for "model learning" assumes the perfect knowledge of the scheduling signal during the system identification process. The questions that we address in this work are: (i) how is the performance of the LPV system identification procedure affected in the presence of such uncertainties? and (ii) how can we improve the LPV system identification when such uncertainties exist? We will examine the first question through simulation studies. Also, to address the latter question, we model such uncertainties in LPV parameters (that we refer to as "error in variables") and include them in the cost function associated with the underlying optimization problem. In conjunction with SVM, the proposed objective function finds the LPV model structure and the corresponding model coefficients in the presence of error in the variables. This is done using the so-called *kernel trick* approach instead of explicitly defining the feature maps (*i.e.*, basis functions) involved [146].

Support vector machines are supervised learning tools originated in modern statistical learning theory that can effectively provide a non-parametric estimation of the dependency structure for linear regression based LPV models [91, 147]. The supervised learning method was originally proposed by [131, 147] to rebuild the inherent functional relationships and structures in the data [22].

This non-parametric functional dependence estimation is more successful in coping with the bias-variance trade-off than semi-parametric approaches like dispersion functions methods [146]. Also, considering l_2 loss functions in the LS-SVM approach gives a variation of the original SVM method that presents an effective model structure learning in the LPV setting. Finding computationally efficient and unique solution of the linear problem are the advantages of these slightly different approaches like LS-SVM over original SVM method. Hence our aim in this work is to employ an effective variation of the LS-SVM method combined with a cost function that focuses not only on prediction error, but also weighs possible uncertainties in the system variables.

Gaussian process (GP) models generalize the Gaussian probability distribution to the function space that is essential for black-box regression problems. They can be seen as the Bayesian version of the well-known support vector machines (SVMs) that provide probabilistic approaches to learning with reproducing kernel Hilbert spaces. Nonparametric Gaussian process models have been widely used in model identification of nonlinear dynamic systems. The predictive performance of GPs has been evaluated in [118] and compared to other modeling approaches like neural networks or local learning methods. In [47], a k-step ahead forecasting of a discrete-time nonlinear and LTI dynamic system is performed using repeated one-step ahead predictions. In the LPV system identification framework, the authors have introduced a Bayesian framework for identification of the coefficients in *finite impulse response* (FIR) dynamic structures in [49]. In this paper, an extension of the standard GP method is formulated to identify the dependency of the LPV model coefficients on the scheduling variables while they are corrupted with a Gaussian noise process.

1.4 Literature survey on Reduced-order Model based Control of Burgers Equation under Model Uncertainties

Computational modeling, simulation and control of nonlinear turbulent systems is a challenging task due to the complexity of the fluid mechanics problems. There are handful of researches targeting the control problem of linear, semi-linear and nonlinear parabolic and hyperbolic partial differential equations (PDEs). In [52], stabilization of a semi-linear parabolic partial differential

equation, in which the heat source depends on the temperature of the whole space, is considered by using boundary control. The adaptive boundary stabilization and control has been investigated for a class of systems described by first-order hyperbolic PDEs with unknown spatially varying parameter in [153]. Also, the predictive control of linear parabolic partial differential equations (PDEs) with state and control constraints was studied in [38]. The Navier-Stokes equation describes many of the underlying phenomena in fluid mechanics. This equation is simplified to Burgers' equation when flow is considered to be incompressible and with the pressure term removed. The Burgers' equation can also be viewed as an intermediate step to capture very critical nonlinear convective behaviors that can model shock waves, some boundary layer problems and traffic flow problems among many others [6, 20].

Over the past three decades, Burgers' equation has been used to gain a better understanding of turbulence and few other complex phenomena in fluid systems. This nonlinear parabolic partial differential equation (PDE) provides a mathematical model that can be used for boundary control and distributed feedback control design purposes. In the present study, we consider this nonlinear PDE with Neumann boundary conditions to develop a reduced order, control-oriented model. To this end, we first approximate this nonlinear PDE with a large number of ordinary differential equations (ODEs) using finite element model (FEM), and then reduce it to the state-space form using proper orthogonal decomposition (POD) method.

The use of reduced-order models can introduce a source of uncertainty imposed by the order of the reduced model. In addition to this, there are other types of uncertainty, i.e., varying parameter, that might affect the accuracy of the extracted reduced model [16, 57, 153]. Hence, the objective is to obtain an accurate reduced-order representation of the original system while ensuring the robustness to uncertainties. An approach based on a dictionary of solutions is developed by [6] as an alternative to using a truncated reduced basis based on proper orthogonal decomposition. The elements of this dictionary are solutions computed for varying values of time and the associated parameter. In [56], a sensitivity analysis is carried out to include the flow and shape parameters influenced during the basis selection process to develop more robust reduced order models for

varying viscosity, changing orientation and shape definition of bodies. In the present research, the model uncertainties are classified in two categories and a robust nonlinear controller is proposed for a trajectory tracking problem. First, the model uncertainties arising from the approximation of the full-order model by the reduced-order one are investigated. Furthermore, the error associated with the varying parameter is studied when the POD basis functions extracted at the nominal parameter ν_0 are used to estimate the full-order model at the new value of the viscosity ν. A nonlinear control strategy based on the reduced-order sliding mode control is then proposed to tackle different kinds of uncertainties arising from parametric and modeling imprecisions in the reduced-order nonlinear model of Burgers' equation.

1.5 Dissertation aims and organization

The aim of this dissertation is to design control algorithm for the deployment of agents on a given environment with an assigned probability distribution function as a measure of uncertainty level in different regions of the environment. The objective is to design a control paradigm that can handle various scenarios where heterogeneous groups of agents can be deployed each of which can handle different assigned tasks. The presented framework can cope with complexities arising from existence of various sources of uncertainty associated every assigned coverage task.

As the next task, we develop new algorithms, using tools from *machine learning* (ML), to address the problems of model reduction, identification and control in the presence of uncertainty. The present task studies the following three problems, first, the system Identification of nonlinear systems using linear parameter varying approach is studied in the presence of uncertainty in scheduling parameters. This is due to the fact that in real world problems the measurement are prone to noise/uncertainty. the reduce-order model based robust nonlinear control of PDE systems are studied for the case where the reduced-model is subjected to model uncertainties. The objective of this work is to design a robust nonlinear control by investigating the different sources of uncertainty and studying the boundedness of the model discrepancies.

The dissertation is organized as follows: Chapter 2 addresses the coverage problem in environments, where several regions of interest exist. To this purpose, a heterogeneous group of robots are deployed to minimize a cost function defined with respect to various spatial probability density functions, each of which describes a desired area for a different group of robots. In Chapter 3, the coverage optimization problem is studied for mobile sensing networks from a team-based perspective. The objective is to locate robots in a given environment so as to minimize the serving cost based on a given density function defining the probability of events in the environment. Chapter 4 defines the main coverage problem as a general optimization over all the teams and their associated agents, where the cost function for each team is defined over the Voronoi of each team with dynamic boundaries. As an application of the proposed team-based method, we study formation control problem within the same framework. Chapter 5 studies the coverage problem in surface flow fields, where it is desired to cover a long region by moving within the boundaries of the flow stream. To this purpose, a group of autonomous mobile sensors are deployed aiming to minimize a sensing cost function. The coverage area, considered to be a region with changing boundaries, is directed to move along the boundaries of the flow until it reaches to the final destination. Chapter 6 introduces a support vector machine approach for system identification of linear parameter varying in presence of error-in-variables (EIV). In conjunction with SVM, the proposed objective function finds the LPV model structure and the corresponding model coefficients in the presence of error in the variables. Chapter 7 proposes a new system identification approach for input-output LPV models is presented in this paper based on Gaussian Process (GP) to compensate for the errors in the scheduling variables. The proposed approach uses a linear approximation to capture the effect of scheduling variables noise on the evaluated coefficient functions on the observed scheduling variables. Chapter 8 focuses on the continuous POD (as opposed to snapshot POD) method and its application for the model order reduction of the forced Burgers' equation, which has characteristics similar to the Navier-Stokes equations. The objective of this work is to take advantage of the underlying characteristics of the continuous POD method to reduce the original model to a number of ODEs that would then be represented in the state-space form. In Chapter 9,

the reduced-order model developed in Chapter 8 is used for the design of a sliding mode controller on the basis of the sliding surfaces defined according to the reduced-order model. Chapter 10 proposes a robust sliding mode controller for dynamic systems governed by Burgers' equation with Neumann boundary conditions in the presence of parameter variations. The main objective is to design a reduced-order model based controller at a nominal value of the system parameter that stabilizes the full order model while being robust with respect to model uncertainties in the obtained reduced-order model. Finally, concluding remarks about the contribution of the dissertation and possible future tracks of research are laid out in Chapter 11.

CHAPTER 2

COVERAGE CONTROL OF MOVING SENSOR NETWORKS IN A COMPLEX ENVIRONMENT WITH MULTIPLE REGIONS OF INTEREST [1]

[1] F. Abbasi, A. Mesbahi and J. Mohammadpour: Coverage Control of Moving Sensor Networks in a Complex Environment with Multiple Regions of Interest. 2016. *Submitted to the American Control Conference.*

ABSTRACT

This paper addresses the coverage problem in environments, where several regions of interest exist. To this purpose, a heterogenous group of robots are deployed to minimize a cost function defined with respect to various spatial probability density functions, each of which describes a desired area for a different group of robots. Each region of interest is assigned to a group of robots with respect to their dynamics and sensing capabilities. A distributed coverage scheme is proposed to allow adjusting to the environment with several important areas in a collaborative way. The regions with higher importance would be covered with a required number of robots. The proposed method also allows for a better allocation of robots to guarantee the desired coverage over the region. Two numerical examples are finally given to examine the proposed coverage approach in case of multiple regions of interest that may need to be covered by certain number of robots.

2.1 INTRODUCTION

Analytical methods have been developed for deployment of a group of robots to perform assigned tasks, such as coverage, in an environment known *a priori*. Typical applications of the coverage problem include search, surveillance, target detection and rescue operations, sensing, and data collection [78,80,81,151]. The major question in coverage problem is how to share the workload in a reliable and efficient way while performing in a distributed way. The distributed method proposed in [32] finds a locally optimal coverage in an environment with a homogenous group of robots based on the Voronoi diagram framework, where each robot implements a control law designed based on the gradient descent method that minimizes the coverage cost in space and time leading to an optimal partitioning.

The underlying optimization problem for coverage control using multiple robots is known to lead to a non-deterministic polynomial-time (NP)-hard problem [46]. Existing methods hence seek for a local solution to this complex optimization problem while ensuring the desired convergence performance. The local solution depends on a given spatial probability density function describing

the likelihood of an event taking place in the environment [99, 124]. The robots concentrate on more important regions in the environment captured by a given density function. If the density function is the sum of two or more probability density functions, the final local solution would then be heavily dependent on the initial positions of the robots. In this case, it is possible that the number of robots assigned to each region does not meet the desired concentration on regions with higher importance. In other words, it is likely that a fewer number of robots are deployed to an area with a wider region of interest which translates to a higher final cost.

A multi-robot system can generally be considered as a homogeneous or heterogeneous system deployed to perform assigned tasks [12, 14, 87]. The generalized Voronoi diagrams depending on a set of weights are used to handle the heterogenous groups of robots as presented in [102, 111, 126]. Furthermore, due to the need for collecting various types of data from a given region, robots may be equipped with different types of sensors to measure the environmental parameters with different rates and precision [53, 87, 141, 143]. It is also possible that a sensor installed on a robot has a particular operational characteristics, and hence, it may be more suitable to be deployed in a certain part of the region. The present work considers the case, where there are more than one important region in the environment, where each important region is assigned to a set of robots that will cover the area using their equipped sensors.

Existing coverage control approaches plan a local optimal path for each robot to address the coverage problem neglecting the possible differences in the dynamics of the robots [116]. For instance, a heterogeneous group of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) were deployed in [50,140] to cover large areas, where it was shown that the dynamical differences may lead to several issues. For example, the UGVs are not able to move as fast as UAVs, but they can be equipped with highly accurate sensors [50, 116]. Hence, it is realistic in many practical applications to take into account the dynamics of the robots in planning the optimal path. This paper also attempts to present a distributed approach that enables the robots to be deployed with respect to their dynamics. The agents with similar dynamics can be assigned to the same region to accomplish the coverage task in a more efficient way. A long-distance important region, with

respect to the initial positions of robots, is assigned to a group of robots that have the dynamic capabilities to move faster than others. For instance, in the UAV/UGV example given above, the UAVs are desired to cover the regions far from the locations they are initially deployed while the UGVs pursue tasks close to their initial deployed position.

A team-based coverage control scheme has been introduced in [2] to facilitate the deployment of a heterogeneous group of robots. The proposed algorithm uses heterogenous teams of robots to handle the coverage of the given environment, but the number of teams and team members are assumed to be independent of the environment. The present work introduces a new coverage control scheme that can allow deploying groups of heterogeneous robots in order to handle the coverage problem in an environment that consists of multiple important regions. Each important region is assigned to a group of robots by taking into account their dynamics and sensing capabilities. To this aim, the density function associated with each robot may differ from its neighbors to reflect the difference in their associated regions of interest. Each robot might team up with others based on its density function, associated dynamics, or sensing characteristics. Therefore, the proposed coverage strategy would make it possible to improve reliability, accuracy, and flexibility of the deployment algorithm by taking into account the differences in the embedded sensors and dynamics of robots through offering an alternative solution method for the underlying sensing cost function. In the proposed formulation, the importance functions associated with the neighboring agents may differ form each other. This implies that, unlike the previous coverage control algorithms, there will be an additional term in the control law that takes into account the potential difference in the importance function of the robots sharing boundaries. Each robot can calculate its Voronoi cell by knowing not only the position of the neighboring robots, but also a data set associated with the difference in their density functions. The required information for calculating the Voronoi cells is obtained through the adjust-communication radius algorithm first developed in [32].

The remainder of this paper is structured as follows. Definitions and the problem statement are provided in Section II. Section III introduces a new optimization problem suited for the coverage

problem in the presence of several regions of interest. The asymptotic convergence of the group of agents imposing the proposed control law is proven through the use of the Barbalat's lemma. Section IV presents simulation results to illustrate the proposed solution method for the coverage problem in the environment with multiple regions of interest with different degrees of importance.

NOTATIONS

We use $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$ to denote the sets of natural, real, and nonnegative real numbers. Also, I_r denotes $r \times r$ identity matrix. We define $\mathcal{Q}$ as a convex polytope in $\mathbb{R}^2$ and let $\mathscr{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_N\}$ be a *partition* of $\mathcal{Q}$ as a collection of closed subsets with disjoint interiors. The boundary of $\mathcal{Q}$ is shown by $\mathcal{Q}^B$. The shared edge of the Voronoi cell V_m with Voronoi cell V_s is shown by V_{ms}^B that is $V_m^B \cap V_s^B$. Moreover, the so-called *distribution density function* (importance function) is denoted by φ, where $\varphi : \mathcal{Q} \to \mathbb{R}_+$ represents the probability of some phenomenon occurring over space $\mathcal{Q}$. The function φ is assumed to be measurable and absolutely continuous. The Euclidean distance function is denoted by $\|\cdot\|$ and $|\mathcal{Q}|$ represents the Lebesgue measure of convex subset $\mathcal{Q}$. With p_i defined as the location of the i^{th} agent, the vector set $\mathcal{P} = (p_1, p_2, \ldots, p_N)$ denotes the location of N agents in the space $\mathcal{Q}$.

2.2 PRELIMINARIES AND PROBLEM STATEMENT

The objective of the coverage related tasks is to ensure a high sensing performance for a mobile network of sensors. This is formulated as an optimization problem that aggregates the sensing performance of all the robots into a so-called sensing cost function. This section gives an overview of the sensing cost function and the necessary modifications to cope with the complexity of the problem at hand.

2.2.1 UNDERLYING OPTIMIZATION PROBLEM

The following locational optimization function has been defined in the literature as a measure of sensing performance

$$\mathcal{H}(P, \mathcal{Q}) = \sum_{i=1}^{N} \int_{\mathcal{W}_i} f(\|q - p_i\|)\varphi(q)\mathrm{d}q, \tag{2.1}$$

where N is the number of agents, P is the set of all agents, and it is assumed that i^{th} agent is assigned to the region $\mathcal{W}_i$. The cost function $\mathcal{H}$ is minimized by finding the optimum locations of the agents and their assigned regions $\mathcal{W}_i$ whose union is $\mathcal{Q}$. As expected, the sensing performance of the agents decays as we move away from their location, and hence, sensing performance can be evaluated as a function of distance from the agent, i.e., $f(\|q - p_i\|)$, where $q \in \mathcal{Q}$. In this context, all the agents are assigned a region over the space assuming that they are homogenous agents equipped with the same sensing devices and pursing the same task over a region. However, due to the possible difference in the agents sensing capabilities, it is likely that the agents can carry different sensors. This is addressed in the present work by introducing an alternative formulation of the coverage problem for heterogenous agents pursuing various tasks in a region characterized by different importance functions.

2.2.2 VORONOI PARTITIONS

This paper aims at addressing the problems of agents deployment and partitioning to efficiently handle multiple tasks. The main objective of this work is to adopt the multi tasking concept in the agents deployment problem and partitioning framework. To do so, we first need to define an optimization problem that can handle the partitioning and deployment within the defined polytope $\mathcal{Q}$. The optimization problem should consider various importance functions representing the difference in the received sensory data or the assigned task in the context of coverage control.

We first partition the polytope $\mathcal{Q}$ into a set of Voronoi cells $\mathcal{V}(\mathcal{P}) = \{V_1, V_2, \ldots, V_N\}$ that are known to provide the optimal partitioning for a set of agents with fixed locations at a given space [32] as

$$V_m = \{q \in \mathcal{Q} | \ \|q - p_m\| \leq \|q - p_r\|, r = 1, \ldots, N, r \neq m\}, \tag{2.2}$$

where p_m denotes the location of m^{th} agent in $\mathcal{Q}$ and $m \in \{1, \ldots, N\}$. The importance function associated with one agent can differ from another; such a difference can well represent heterogeneity in their embedded sensing devices or assigned tasks in the distributed system of agents. We recall the basic characteristics of the Voronoi partitions including their associated mass, centroid, and polar moment of inertia defined as [32].

$$\begin{aligned} M_{V_m} = \int_{V_m} \varphi_m(q)\mathrm{d}q, \quad C_{V_m} = \frac{1}{M_{V_m}} \int_{V_m} q\ \varphi_m(q)\mathrm{d}q, \\ J_{V_m,p_m} = \int_{V_m} \|q - p_m\|^2 \varphi_m(q)\mathrm{d}q, \end{aligned} \tag{2.3}$$

where $\varphi_m(q)$ represents the importance function associated with m^{th} agent.

2.3 Coverage Control in an Environment with Multiple Regions of Interest

The deployment task of interest in this paper can be addressed by solving an optimization problem that represents the agents with various regions of interest collectively. The optimization problem seeks to achieve the optimal deployment over the convex polytope $\mathcal{Q}$, where each agent's region needs to be determined. Hence, the following cost function is defined

$$\mathcal{G}(\mathcal{P}, \mathscr{Q}) = \sum_{m=1}^{N} \int_{\mathcal{Q}_m} \|q - p_m\|^2 \varphi_m(q)\mathrm{d}q, \tag{2.4}$$

in which the sensing performance is considered as $f(\|q - p_m\|) = \|q - p_m\|^2$ for the m^{th} agent with the importance function of φ_m. The solution to minimizing $\mathcal{G}$ in (2.4) gives a local minimum to the deployment problem, where agents are collaborating and heterogenous. It can be shown that among different partitioning schemes, the Voronoi partitions are optimum in the sense of minimizing defined cost function (2.4) [32]. Hence, for a given set of agents position $\mathcal{P} \in \mathcal{Q}$ and a *partition* $\mathscr{Q}$ of $\mathcal{Q}$, it satisfies

$$\mathcal{G}(\mathcal{P}, \mathcal{V}(\mathcal{P})) \leq \mathcal{G}(\mathcal{P}, \mathscr{Q}), \tag{2.5}$$

which implies that the Voronoi cells represent the optimum partitioning. To obtain the optimum configuration of the agents, we need to solve the problem of minimizing (2.4). In order to avoid

the complexity involved in obtaining the global minimum of an NP-hard (Non-deterministic Polynomial-time hard) problem, the local minimum of the cost function is sought for by taking the derivative of the sensing cost function with respect to the agents position as

$$\frac{\partial \mathcal{G}}{\partial p_m} = \frac{\partial}{\partial p_m} \sum_{r=1}^{N} \int_{V_r} \|q - p_r\|^2 \varphi_r(q) \mathrm{d}q, \quad m = 1, \ldots, N. \tag{2.6}$$

The solution to this problem differs from the conventional sensing cost functions due to various importance functions involved that represent heterogeneity. The derivative with respect to the coordinates of agent p_m is obtained as

$$\frac{\partial \mathcal{G}}{\partial p_m} = \int_{V_m} \frac{\partial}{\partial p_m} \|q - p_m\|^2 \varphi_m(q) \mathrm{d}q + \sum_{r=1}^{N} \int_{V_r^B} \|q - p_r\|^2 \varphi_r(q) \frac{\partial V_r^B}{\partial p_m} \mathcal{N}_r \mathrm{d}q, \tag{2.7}$$

where V_r^B represents the boundary of the Voronoi cell V_r. As it can be inferred from the definition of the Voronoi partitioning, the boundary of the Voronoi cell V_r that is in the neighborhood of m^{th} robot is dependent on p_m.

Remark 2.1 *The agents whose Voronoi cells do not share any edges with the Voronoi cell associated with p_m are independent of p_m. This implies that $\frac{\partial V_r^B}{\partial p_m} = 0$ for any $r = 1, \ldots, N$ that $p_r \notin \mathcal{N}_{p_m}$ where $\mathcal{N}_{p_m}$ represents the set of agents that share boundaries with m^{th} agent.*

Remark 2.2 *The integral on each boundary shared with neighboring agents is the same for agents on both sides except that the normals have opposite signs, i.e., $\mathcal{N}_{sm} = -\mathcal{N}_{ms}$, where $\mathcal{N}_{ms}$ is the normal vector for the edge of the Voronoi cell V_m, i.e., V_{ms}^B, that is shared with another Voronoi cell V_s.*

The last term in (2.7) can be rewritten as follows

$$\sum_{r=1}^{N} \int_{V_r^B} \|q - p_r\|^2 \varphi_r(q) \frac{\partial V_r^B}{\partial p_m} \mathcal{N}_r \mathrm{d}q = \sum_{p_r \in \mathcal{N}_{p_m}} \int_{V_r^B} \|q - p_r\|^2 \varphi_r(q) \frac{\partial V_r^B}{\partial p_m} \mathcal{N}_r \mathrm{d}q + \sum_{p_r \notin \mathcal{N}_{p_m}} \int_{V_r^B} \|q - p_r\|^2 \varphi_r(q) \frac{\partial V_r^B}{\partial p_m} \mathcal{N}_r \mathrm{d}q. \tag{2.8}$$

According to Remark 2.1, the last term is reduced to the integral on the boundary of the agent m^{th} agent. Then, we obtain the following

$$\sum_{r=1}^{N}\int_{V_r^B}\|q-p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r\mathrm{d}q=\sum_{p_r\in\mathcal{N}_{p_m}}\int_{V_r^B}\|q-p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r\mathrm{d}q+$$
$$\int_{V_m^B}\|q-p_m\|^2\varphi_m(q)\frac{\partial V_m^B}{\partial p_m}\mathcal{N}_m\mathrm{d}q=\sum_{p_r\in\mathcal{N}_{p_m}}\int_{V_{rm}^B}\|q-p_r\|^2\varphi_r(q)\frac{\partial V_{rm}^B}{\partial p_m}\mathcal{N}_{rm}\mathrm{d}q+$$
$$\sum_{p_r\in\mathcal{N}_{p_m}}\int_{V_{mr}^B}\|q-p_m\|^2\varphi_m(q)\frac{\partial V_{mr}^B}{\partial p_m}\mathcal{N}_{mr}\mathrm{d}q. \quad (2.9)$$

Using Remark 2.2 and the following equality

$$\frac{\partial V_{mr}^B}{\partial p_m}=\frac{\partial V_{rm}^B}{\partial p_m}, \quad (2.10)$$

it can be concluded that

$$\sum_{r=1}^{N}\int_{V_r^B}\|q-p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r\mathrm{d}q=\sum_{p_r\in\mathcal{N}_{p_m}}\int_{V_{mr}^B}(\|q-p_m\|^2\varphi_m(q)-$$
$$\|q-p_r\|^2\varphi_r(q))\frac{\partial V_{mr}^B}{\partial p_m}\mathcal{N}_{mr}\mathrm{d}q. \quad (2.11)$$

From the definition of the Voronoi cell (2.2), we have $\|q-p_m\|=\|q-p_r\|$ for any $r\in\mathcal{N}_{p_m}$. Hence, the following is obtained

$$\sum_{r=1}^{N}\int_{V_r^B}\|q-p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r\mathrm{d}q=\sum_{p_r\in\mathcal{N}_{p_m}}\int_{V_{mr}^B}\|q-p_m\|^2\big(\varphi_m(q)-\varphi_r(q)\big)\frac{\partial V_{mr}^B}{\partial p_m}\mathcal{N}_{mr}\mathrm{d}q. \quad (2.12)$$

It is noted that each agent might also be the neighbor of other agents that pursue a similar task through the same importance function. If that is the case, then the result of (2.12) is equal to zero. However, this is not the case in general and hence, we need to evaluate the integral on the given boundaries.

The line to which the points on the agents boundaries belong can be described by

$$(\mathcal{N}_{mr})^\top(q-\frac{p_r+p_m}{2})=0,\quad q\in V_{mr}^B, \quad (2.13)$$

where V_{mr}^B is the shared boundary between agents m and r. The normal vector $\mathcal{N}_{mr}$ associated with V_{mr}^B is obtained by

$$\mathcal{N}_{mr}=\frac{p_r-p_m}{\|p_r-p_m\|}. \quad (2.14)$$

The partial derivative of (2.13) with respect to p_m is obtained as follows

$$\frac{\partial \mathcal{N}_{mr}}{\partial p_m}(q - \frac{p_r + p_m}{2}) + (\frac{\partial V_{mr}^B}{\partial p_m} - \frac{1}{2})\mathcal{N}_{mr} = 0, \tag{2.15}$$

where

$$\frac{\partial \mathcal{N}_{mr}}{\partial p_m} = \frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}. \tag{2.16}$$

By substituting (2.16) into (2.15), we obtain

$$\frac{\partial V_{mr}^B}{\partial p_m}\mathcal{N}_{mr} = \frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}(\frac{p_r + p_m}{2} - q) + \frac{1}{2}\mathcal{N}_{mr}, \quad q \in V_{mr}^B. \tag{2.17}$$

Substituting (2.17) back into (2.12) leads to

$$\sum_{r=1}^{N}\int_{V_r^B}\|q - p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r \mathrm{d}q = \sum_{p_r \in \mathcal{N}_{p_m}}\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}\int_{V_{mr}^B}\Big(\|q - p_m\|^2(\varphi_m(q) - \varphi_r(q))\frac{p_r + p_m}{2} - \|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)q\Big)\mathrm{d}q + \frac{1}{2}\mathcal{N}_{mr}\int_{V_{mr}^B}\|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)\mathrm{d}q. \tag{2.18}$$

By rearranging the terms in (2.18), we obtain

$$\sum_{r=1}^{N}\int_{V_r^B}\|q - p_r\|^2\varphi_r(q)\frac{\partial V_r^B}{\partial p_m}\mathcal{N}_r \mathrm{d}q = \sum_{p_r \in \mathcal{N}_{p_m}}\left(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}\frac{p_r + p_m}{2} + \frac{1}{2}\mathcal{N}_{mr}\right)\int_{V_{mr}^B}\|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)\mathrm{d}q - \frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}\int_{V_{mr}^B}\|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)q\mathrm{d}q. \tag{2.19}$$

The derivative (2.7) can then be rewritten as

$$\frac{\partial \mathcal{G}}{\partial p_m} = \int_{V_m}\frac{\partial}{\partial p_m}\|q - p_m\|^2\varphi_m(q)\mathrm{d}q + \sum_{p_r \in \mathcal{N}_{p_m}}\left(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}\frac{p_r + p_m}{2} + \frac{1}{2}\mathcal{N}_{mr}\right)\int_{V_{mr}^B}\|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)\mathrm{d}q - \frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|}\int_{V_{mr}^B}\|q - p_m\|^2\big(\varphi_m(q) - \varphi_r(q)\big)q\mathrm{d}q. \tag{2.20}$$

Further simplification of (2.20) using the equations of mass and centroid (2.3) results in

$$\frac{\partial \mathcal{G}}{\partial p_m} = -2M_{V_m}(C_{V_m} - p_m) + \sum_{p_r \in \mathcal{N}_{p_m}} \left(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \frac{p_r + p_m}{2} + \frac{1}{2}\mathcal{N}_{mr} \right)$$
$$\int_{V_{mr}^B} \|q - p_m\|^2 \big(\varphi_m(q) - \varphi_r(q)\big) \mathrm{d}q - \frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \int_{V_{mr}^B} \|q - p_m\|^2 \big(\varphi_m(q) - \varphi_r(q)\big) q \mathrm{d}q. \tag{2.21}$$

Considering the following notations

$$\mathcal{G}_{mr} = \int_{V_{mr}^B} \|q - p_m\|^2 \big(\varphi_m(q) - \varphi_r(q)\big) \mathrm{d}q, \tag{2.22}$$

$$\mathcal{L}_{mr} = \int_{V_{mr}^B} \|q - p_m\|^2 (\varphi_m(q) - \varphi_r(q)) q \mathrm{d}q, \tag{2.23}$$

equation (2.21) can be written as

$$\frac{\partial \mathcal{G}}{\partial p_m} = -2M_{V_m}(C_{V_m} - p_m) + \sum_{p_r \in \mathcal{N}_{p_m}} \big(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \frac{p_r + p_m}{2} + \frac{1}{2}\mathcal{N}_{mr} \big) \mathcal{G}_{mr} -$$
$$\big(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \big) \mathcal{L}_{mr}. \tag{2.24}$$

As observed from (2.24), we need to obtain the integral on the boundaries of the agents' Voronoi for $\mathcal{G}_{mr}$ and $\mathcal{L}_{mr}$ that take into account the effect of the difference in the importance function of the neighboring agents.

2.3.1 CONTROLLER DESIGN

The gradient-decent based control law is utilized here to guarantee the convergence of the agents to their equilibrium point while pursing various objectives. The following dynamics is imposed on each agent

$$\dot{p}_m = u_m = \frac{K_m}{2M_{V_m}} \left(- \frac{\partial \mathcal{G}}{\partial p_m} \right) = \frac{K_m}{2M_{V_m}} \Big(2M_{V_m}(C_{V_m} - p_m) - \gamma_m \Big), \quad m = 1, \ldots, N, \tag{2.25}$$

where K_m is a positive scalar and

$$\gamma_m = \sum_{p_r \in \mathcal{N}_{p_m}} \big(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \frac{p_r + p_m}{2} + \frac{1}{2}\mathcal{N}_{mr} \big) \mathcal{G}_{mr} - \big(\frac{\mathcal{N}_{mr}(\mathcal{N}_{mr})^\top - I_2}{\|p_r - p_m\|} \big) \mathcal{L}_{mr}. \tag{2.26}$$

By dividing K_m by varying $2M_{V_m}$, the control input is normalized to distribute the effect of both terms in the controller design. As expected, the first term drives the agent towards its centroid while the second term is associated with various regions of interest when multiple agents collaborate. In other words, the above control law ensures that the agents move to their optimum location while taking into account the difference in importance functions.

2.3.2 Computation of the Voronoi cells

The Voronoi cells associated with each agent require a set of information to be computed over time. Based on the algorithm presented in [32], the agents are able to compute their Voronoi cells by communicating with the neighboring agents. In order to obtain the control law for each agent in the proposed approach of this paper, the agents need to receive the integral of $\|q - p_r\|\varphi_r(q)$ and $\|q - p_r\|\varphi_r(q)q$ on the edge shared with other agents from the neighboring agents. Even though this increases the amount of data to be exchanged through the communication link, however, it eliminates the need for equipping all the agents with the same sensing devices in practical applications. In fact, the proposed distributed approach enables a group of heterogenous agents capable of accomplishing the coverage task while acquiring certain information from their neighbors.

2.3.3 Convergence of the proposed controller

The proposed controller drives the agents to their centroid while taking the heterogeneity of the agents into account. To ensure the convergence of the agents to their collective local optimum, the following lemma is proposed.

Lemma 2.3.1 *The agents converge to a local minimum by imposing the control law (2.25). That is,*

$$\lim_{\tau\to\infty} \left\| -2M_{V_m}(\tau)(C_{V_m}(\tau) - p_m(\tau)) + \gamma_m(\tau) \right\| = 0, \tag{2.27}$$

for $\forall m \in \{1, \ldots, N\}$.

Proof 1 *The asymptotic behavior cannot be proved through invoking the invariant set theorem for time-varying systems; we hence use the Barbalat's lemma to prove the asymptotic convergence of the group of agents when each agent follows an optimal configuration. To this aim, the Lyapunov-like function associated with each team is defined as*

$$V = \mathcal{G}(\cdot, \cdot). \tag{2.28}$$

The derivative of this function is obtained as

$$\dot{V} = \sum_{m=1}^{N} (\frac{\partial p_m}{\partial \tau})^\top \frac{\partial \mathcal{G}}{\partial p_m}. \tag{2.29}$$

Substituting (2.21) and (2.25) into (2.29), we obtain

$$\dot{V} = -\sum_{m=1}^{N} \frac{K_m}{2M_{V_m}} \Big(-2M_{V_m}(C_{V_m} - p_m) + \gamma_m \Big)^\top \Big(-2M_{V_m}(C_{V_m} - p_m) + \gamma_m \Big). \tag{2.30}$$

Since M_{V_m} and K_m are positive scalars, it can be concluded that the derivative (2.30) is non-positive, $\dot{V} \leq 0$. Due to the positivity of the cost function $\mathcal{G}$, it is also concluded that the Lyapunov-like function (2.28) is non increasing and hence lower bounded. As shown in [124], $\ddot{V}(\tau)$ is uniformly bounded that results in uniform continuity of the $\dot{V}(\tau)$. Now, due to the boundedness of $V(\tau)$ and the continuity of $\dot{V}(\tau)$, it is concluded by Barbalat's lemma that

$$\lim_{\tau \to \infty} \dot{V}(\tau) = 0 \quad \Longrightarrow \quad \lim_{\tau \to \infty} \| -2M_{V_m}(\tau)(C_{V_m}(\tau) - p_m(\tau)) + \gamma_m(\tau) \| = 0, \tag{2.31}$$

for $\forall m \in \{1, \ldots, N\}$.

2.4 Simulation Results and Discussion

Simulation results are shown here to examine the efficiency of the proposed coverage control approach. The objective is to determine the coverage configuration and the path taken by each robot to yield the optimal deployment. In the first scenario, three different importance functions are assigned to three groups, each composed of six robots. As an initial configuration, the robots are deployed randomly to accomplish their associated coverage task while communicating with agents

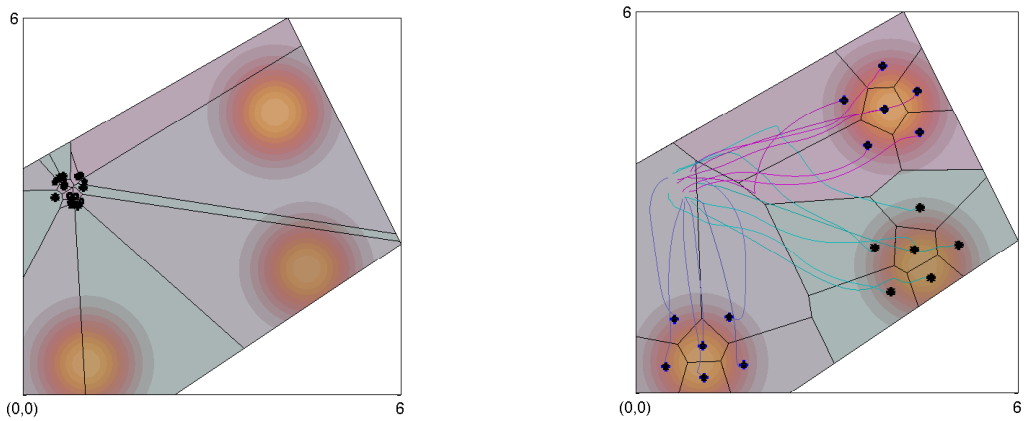


Figure 2.1: Initial (top) and final configurations (bottom) for three groups of six agents, each of which pursues a different coverage task.

pursuing a different task. This illustrates the heterogeneity of the agents in terms of the underlying coverage density function that represents the potential differences in the dynamics or sensing capabilities of the robots. As shown in Figure 2.1, the robots collaborate to ultimately converge to their assigned region of interest. It is noted that the controller gain is chosen to be $K_m = 0.073$ for the first example. Furthermore, the convergence of the sensing cost function is shown in Figure 2.2. As the second scenario, three groups composed of 8, 4, and 6 robots are deployed to cover a given region. The goal here is to demonstrate how the proposed coverage control method works for there multiple regions of interest with various degrees of importance (various values of the variance). Hence, the proposed method enables us to deploy different number of robots to each region with respect to the complexity of the underlying sensing or coverage tasks. The control gain for this scenario is chosen to be $K_m = 0.012$. It is also noted that throughout this work, it is assumed that the importance functions are known *a priori*. As the simulation results show, the ultimate goal is achieved, where the larger regions are assigned to a higher number of agents or the agents with better sensing performance are given a larger region to cover. The convergence of the sensing cost function for the second scenario is shown in Figure 2.4.

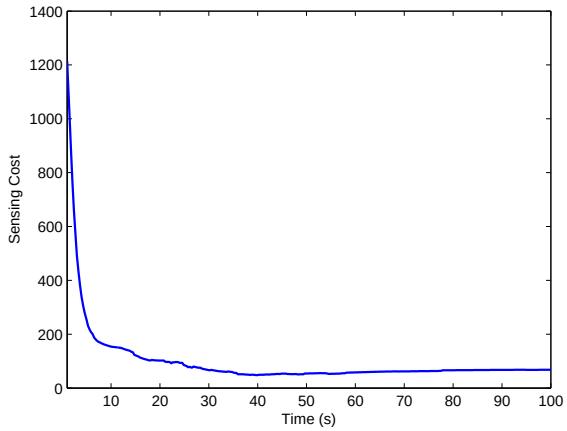


Figure 2.2: Convergence of the sensing cost for the first scenario.

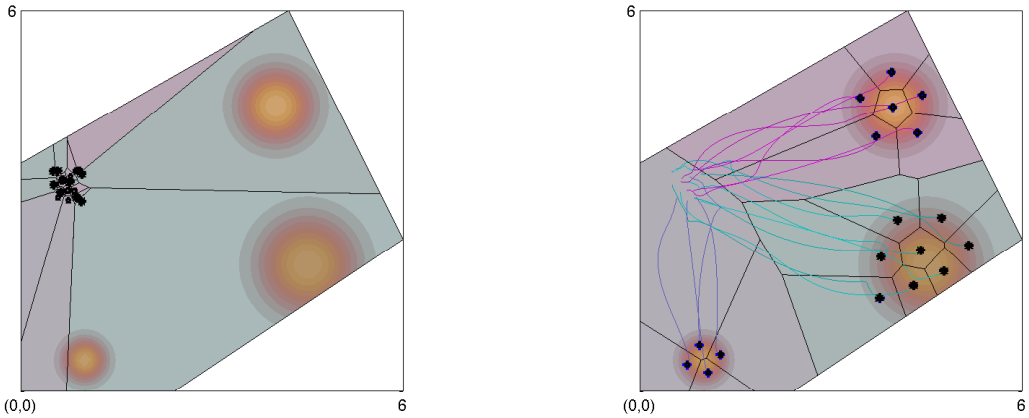


Figure 2.3: Initial (top) and final configurations (bottom) for three groups of 8, 4, and 6 agents, each of which is deployed to cover a region of interest. The results show the flexibility of the proposed approach to cope with problems, where there is a need to assign various number of robots to different regions.

2.5 Concluding Remarks

In this paper, a coverage control approach is proposed for adapting to environments with several important regions. The proposed algorithm enables the deployment of a group of heterogeneous

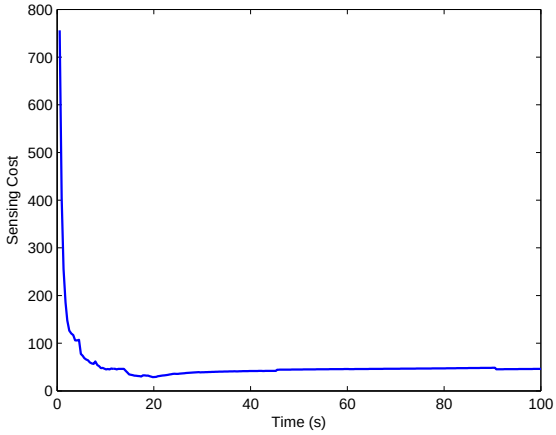


Figure 2.4: Convergence of the sensing cost for the second scenario.

robots with potentially different dynamics and sensing capabilities to perform a desired task while minimizing the sensing cost. Each region of interest is allocated to a group of agents based on the similarity in sensing capabilities and dynamics. Also, the number of robots that need to be deployed over the regions of interest is decided based on the number of importance functions and their associated degree of importance. Robots can compute their Voronoi cells relying on a distributed communication algorithm with their neighbor robots. The proposed approach is an attempt to modify and improve existing methods in a way that several groups of heterogeneous robots can divide the region among themselves based on their capabilities.

CHAPTER 3

TEAM-BASED COVERAGE CONTROL OF MOVING SENSORS NETWORK [1]

[1] F. Abbasi, A. Mesbahi, J. Mohammadpour: Team-based Coverage Control of Moving Sensors Network. 2016. In *Proc. of the American Control Conference*, Boston, MA: pp. 5691-5696.

ABSTRACT

In this paper, the coverage optimization problem is studied for mobile sensing networks from a team-based perspective. The objective is to locate robots in a given environment so as to minimize the serving cost based on a given density function defining the probability of events in the environment. A team concept is introduced here to allow adjusting to the changes in the environment or the assigned task in a collaborative way as a team. Firstly, an optimization problem is solved in the team level in order to assign a sub-region to each team. The region assigned to each team is then divided among its members to minimize a locational cost. A distributed communication algorithm is also given, in which each team member, e.g., robot can access the information of its neighbors in order to compute the associated region. Finally, a numerical simulation is given to demonstrate the effectiveness of the presented approach. The results indicate that agents in both teams and agents level try to minimize the cost function in order to reach the optimal configuration considering the given density function.

3.1 INTRODUCTION

There have been advances in the deployment techniques for a group of robots in a given environment to perform the assigned distributed tasks examples of such tasks include surveillance, search and rescue operations, sensing, and data collection [78, 85, 151]. The core problem can be seen as a work load sharing task to assign the share of each agent from the total work load. Assuming that the agents are identical, the partitioning task is carried out to divide the space among the agents equally in space and time leading to an optimal partitioning [14, 101]. The underlying optimization problem is a NP-hard (Non-deterministic Polynomial-time hard) [46], and hence, local minimization of the cost function is desired. This is the major goal behind the controller design that is discussed later.

The concept of Voronoi partition and generalized Voronoi partitions plays a fundamental role in the optimal space partitioning problem. Among different feasible type of partitionings, the Voronoi

cells are considered as the optimal partitioning for the space sharing tasks [32]. Depending on the way the partitioning is defined, there are various kinds of Voronoi cells like power diagrams, additively-weighted and multiplicatively-weighted Voronoi partitions as introduced in [11, 126].

Due to the advantages of the distributed systems such as reliability, speed and economics over the centralized systems, the distributed deployment algorithms have been proposed for the work load sharing and partitioning tasks. To this aim, each robot should only exchange information collected via its sensors, e.g., position and velocity, with other agents and negotiate its scheduled task with a number of other agents. Then, each robot locally generates an appropriate control action using the gathered information from its neighbors. For mobile sensing networks, a distributed control strategy has been presented in [32] to equally divide an assigned area into subregions where each robot is able to obtain locations of its neighbors through the adjust-communication-radius algorithm. Then, it computes the associated Voronoi cell and moves toward the centroid of its Voronoi cell based on the gradient descent method that gives the optimal solution for equal partitioning problem [37].

The defined objective function represents the performance measure of the agents. In essence, the solution to the optimization problem maximizes the performance index in a way that the agents are deployed in the most optimum way. In other words, the distance from each agent to the points inside its allocated area is minimized in the coverage problem that results in a higher sensing performance for each agent [32]. This algorithm has been extended for power diagrams so that not only equitable partitions can be obtained in a spatially distributed manner, but also equitable and median Voronoi diagrams are acquired [111].

The existing approaches on the coverage control are based on the assumption that all robots belong to one team. However, this assumption is not realistic in many real-world applications, as the agents may differ from dynamics and communication perspective. A multi-robot system can generally be considered as a homogeneous or heterogeneous system that translates to deploying a variety of robots with respect to the assigned task. In the present work, an alternative coverage strategy is presented that aims at taking the differences in the robots dynamics into account by

proposing the team concept. Each robot might team up with other ones based on its assigned task, associated dynamics or embedded communication system. Therefore, it would be possible to increase reliability and flexibility of the deployment algorithm.

The present work introduces a new team-based coverage control scheme which can handle different scenarios in heterogeneous systems of robots. The presented approach addresses the problem of the agents deployment by considering teams of robots instead of evaluating each agent individually. The main problem can be defined as a two level optimization problem, where one is defined inside each team and another optimization is defined among the teams in the overall coverage space. The agents where will keep move towards the local minima until the optimum configuration is achieved. In the proposed approach, firstly, a local minimum to the deployment problem is obtained in the team level. Then, a second optimization problem is solved to guarantee the convergence of the agents to their optimum location generating their respective Voronoi cells inside the teams. By considering the nucleus as the associated Voronoi centroid of each team, the optimization problem is defined at the team level by maximizing the performance of the team.

The agents in each team can be classified into two groups of interior members and members on the boundaries based on whether they share a boundary with the agents belonging to other teams or not. Each interior member is able to compute its own Voronoi cell by only knowing the location of its neighbors. The location of the neighbors are obtained by using a modification of the adjust-communication-radius algorithm which has been presented in [32]. However, this does not hold for the members on the boundaries. These members need not only the location of neighbor robots in their associated team, but also the nucleuses of the neighboring teams with their team Voronoi cell. The agents on the boundaries can obtain the nuclei of neighboring teams by the proposed communication algorithm in Section IV.

The remainder of this paper is structured as follows. Definitions and the problem statement are provided in Section II. Section III introduces a two-level optimization problem. An extended version of the classic Lloyd algorithm is applied to ensure the coverage properties. The agents com-

munication characteristics and protocols are discussed in Section IV. Section V presents numerical simulation results to illustrate the team-based partitioning.

3.1.1 NOTATIONS

We use $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_+$ to denote the sets of positive natural, real, and nonnegative real numbers. The closed circle centered at $c \in \mathbb{R}^2$ with radius $r \in \mathbb{R}_+$ is defined by $B(c, r) := \{x \in \mathbb{R}^2 | \|x - c\| \leq r\}$. We define $\mathcal{Q}$ as a convex polytope in $\mathbb{R}^2$ and let $\mathscr{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_t\}$ be a partition of $\mathcal{Q}$ as a collection of t closed subsets with disjoint interiors. Moreover, the so-called *distribution density function* is denoted by φ where $\varphi : \mathcal{Q} \to \mathbb{R}_+$ represents the probability of some phenomenon happening over space $\mathcal{Q}$. The function φ is assumed to be measurable and absolutely continuous. The Euclidean distance function is denoted by $\|\cdot\|$ and $|\mathcal{Q}|$ represents the Lebesgue measure of convex subset $\mathcal{Q}$. The vector set $\mathcal{P}_t = (p_{t1}, p_{t2}, \ldots, p_{tn_t})$ is the location of n_t agents belonging to t^{th} team moving in the space $\mathcal{Q}_t$. As expected, the sensing performance of the agents decay as we move away from their location, and hence, sensing performance can be evaluated as a function of distance from the agent, i.e., $f(\|q - p_{tm}\|)$ where $q \in \mathcal{Q}$.

3.2 A TEAM-BASED OPTIMIZATION SCHEME

3.2.1 OPTIMIZATION PROBLEM

In the literature, the locational optimization function is presented in the following form that is translated to maximizing the sensing performance.

$$\mathcal{H}(P, \mathcal{Q}) = \sum_{i=1}^{N} \int_{\mathcal{Q}_i} f(\|q - p_i\|)\varphi(q)\mathrm{d}q, \tag{3.1}$$

where for n number of teams, $N = \sum_{t=1}^{n} n_t$, and P is the set of all agents. It is assumed that i^{th} agent is assigned to the region $\mathcal{Q}_i$ and the cost function $\mathcal{H}$ is minimized by finding the optimum locations of the agents and their assigned regions $\mathcal{Q}_i$. In this context, all the agents are assigned over the space $\mathcal{Q}$ no matter how the agents can collaborate or coordinate with each other in more local

platform. This is addressed in the present work by introducing a team-based partition of the agents that considers agents as a collection of multiple team pursuing their assigned task or objective.

3.2.2 VORONOI PARTITIONS

The core objective is to adopt the team concept in the agents deployment and partitioning framework. To do so, we first need to define an optimization problem that can handle not only the deployment partitioning and tasks inside teams but also the partitioning inside the defined polytope $\mathcal{Q}$. This can be done by breaking the optimization problem into two interconnected functions in a way that the solution to each problem represents the optimum configuration of the teams and their associated agents.

To start with, we define the set of teams by $\mathcal{L} = (l_1, l_2, \ldots, l_n)$ where each $l_t, t = 1 : n$, represents the nucleus of team t that is a function of the agents position in the associated team $l_t = g(p_{t1}, p_{t2}, \ldots, p_{tn_t})$. The function dependency of l_t on the position of the agents is discussed later. Now, we can partition the polytope $\mathcal{Q}$ into a set of Voronoi cells $\mathcal{V}(\mathcal{L}) = \{V_1, V_2, \ldots, V_n\}$ considered as the optimal partitioning for a fixed set of agents location at a given space as

$$V_t = \{q \in \mathcal{Q} |\ \|q - l_t\| \leq \|q - l_s\|\}. \tag{3.2}$$

The obtained Voronoi cells associated with the nuclei of the teams are then considered as the convex polytopes set to deploy their associated agents. Therefore, the sub-partitions are defined on the basis of the Voronoi cells V_t obtained from the team level partitioning. The Voronoi partitions $\mathcal{V}_t(\mathcal{P}_t) = \{V_{t1}, V_{t2}, \ldots, V_{tn_t}\}$ generated by the agents $(p_{t1}, p_{t2}, \ldots, p_{tn_t})$ belonging to t^{th} team are defined as

$$V_{tm} = \{q \in V_t |\ \|q - p_{tm}\| \leq \|q - p_{tr}\|\}, \tag{3.3}$$

where p_{tm} denotes location of m^{th} agent in t^{th} team such that $m \in \{1, \ldots, n_t\}$.

We recall the basic characteristics of the Voronoi partitions including their associated mass, centroid and polar moment of inertia as [32]

$$\begin{aligned} M_{V_{tm}} = \int_{V_{tm}} \varphi(q)\mathrm{d}q,\ C_{V_{tm}} = \frac{1}{M_{V_{tm}}} \int_{V_{tm}} q\ \varphi(q)\mathrm{d}q, \\ J_{V_{tm},p_{tm}} = \int_{V_{tm}} \|q - p_{tm}\|^2 \varphi(q)\mathrm{d}q. \end{aligned} \tag{3.4}$$

In addition to the defined parameters for the Voronoi cells of each agent inside the teams, we also need to define the characteristics of the teams's Voronoi cells. According to the definition of the Voronoi partitions, it can be deduced that

$$M_{V_t} = \sum_{m=1}^{n_t} M_{V_{tm}},\ C_{V_t} = \frac{1}{M_{V_t}} \int_{V_t} q\ \varphi(q)\mathrm{d}q. \tag{3.5}$$

The nucleus of the team that is a function of the agents position is defined as

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} p_{tm}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{3.6}$$

As described earlier, the nucleus is a representative of the agents position in the team and can be considered as the collective position of the agents for drawing the Voronoi diagram of the teams V_t. The agents formation is called centroidal Voronoi configuration if it satisfies $l_t = C_{V_t}$, $p_{tm} = C_{V_{tm}}$ for both interior agents and nuclei of the teams.

3.2.3 LOCAL OPTIMIZATION PROBLEM

The deployment task in the presented team-based scheme can be addressed by solving a two level optimization problem. At first, we need to define an optimization problem that can guarantee the optimum configuration and partitioning of the convex polytope $\mathcal{Q}$ while applying the nuclei of the teams as functions of the agent position. Hence, the cost function for the nuclei of the teams is written in the following form

$$\mathcal{G}(\mathcal{L}, \mathcal{Q}) = \sum_{t=1}^{n} \int_{\mathcal{Q}_t} \|q - l_t\|^2 \varphi(q)\mathrm{d}q, \tag{3.7}$$

where the sensing performance is considered as $f(\|q - l_t\|) = \|q - l_t\|^2$. The solution to (3.7) gives the local minimum to the deployment problem in the teams level. Once the optimum partitioning

is achieved at the teams level, we need to define a second optimization problem to guarantee the convergence of the agents inside the teams to their optimum location generating their respective Voronoi cells. We define a set of polygons $\mathscr{Q}_t = \{Q_{t1}, Q_{t2}, \ldots, Q_{tn_t}\}$ with disjoint interiors whose union is V_t. The following cost function is defined for each team as a function of the position of its associated agents

$$\mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t) = \sum_{m=1}^{n_t} \int_{Q_{tm}} \|q - p_{tm}\|^2 \varphi(q) \mathrm{d}q. \tag{3.8}$$

It is proven that among different partitioning schemes the Voronoi partitions are optimum in the sense of minimizing both individually defined cost functions (3.7) and (3.8) [19,1 in Nowzari work]. Hence, for a given set of nuclei $\mathcal{L} \in \mathcal{Q}$, agents position $\mathcal{P}_t \in V_t$, a partition $\mathscr{Q}$ of $\mathcal{Q}$ and a partition $\mathscr{Q}_t$ of V_t, it satisfies

$$\mathcal{G}(\mathcal{L}, \mathcal{V}(\mathcal{L})) \leq \mathcal{G}(\mathcal{L}, \mathscr{Q}), \tag{3.9}$$

$$\mathcal{G}_t(\mathcal{P}_t, \mathcal{V}_t(\mathcal{P}_t)) \leq \mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t). \tag{3.10}$$

This concludes that the Voronoi cells represent the optimum partitioning. Furthermore, for any $\mathcal{L}' = (l'_1, l'_2, \ldots, l'_n) \in \mathcal{Q}$ and $\mathcal{P}'_t = (p'_{t1}, p'_{t2}, \ldots, p'_{tn_t}) \in V_t$ satisfying $\|l'_t - C_{V_t}\| \leq \|l_t - C_{V_t}\|$ and $\|p'_{tm} - C_{V_{tm}}\| \leq \|p_{tm} - C_{V_{tm}}\|$, respectively, we have

$$\mathcal{G}(\mathcal{L}', \mathscr{Q}) \leq \mathcal{G}(\mathcal{L}, \mathscr{Q}), \tag{3.11}$$

$$\mathcal{G}_t(\mathcal{P}'_t, \mathscr{Q}_t) \leq \mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t). \tag{3.12}$$

In other words, the given cost function is minimized when agents are at the centroids of their corresponding Voronoi cells.

3.3 A Two-Step Optimization Problem

The problem of deploying teams of robots can be broken into two optimization problems. The first function to be minimized represents the cost function associated with partitioning the main space into partitions related to the teams of agents. This is followed by another optimization problem that is solved inside each team. The solution to the second optimization problem result in deploying

the agents in an optimum way inside the teams. To achieve this, we assume that both functions $\mathcal{G}$ and $\mathcal{G}_t$ in (3.7) and (3.8) are smooth and continuous over their regions. Then, the derivatives of the cost functions are obtained as [37]

$$\frac{\partial \mathcal{G}}{\partial l_t} = M_{V_t}(l_t - C_{V_t}), \tag{3.13}$$

$$\frac{\partial \mathcal{G}_t}{\partial p_{tm}} = M_{V_{tm}}(p_{tm} - C_{V_{tm}}). \tag{3.14}$$

It can be seen that if the teams nucleus and agents position move to the centroid of their Voronoi cells, the local minimum is achieved. Accordingly, the critical partitions and points for $\mathcal{G}$ and $\mathcal{G}_t$ are called centroidal Voronoi partitions.

3.3.1 Equivalence of agents and teams level optimization

As discussed earlier, we need to first assign the optimum partitions to the teams of agents and then deploy the robots in the assigned regions to their associated teams. In this section, we show that these two optimization problems are interrelated. In fact, it is shown that the solution to the second optimization problem can lead to the solution to the first problem considering the given dependence of the nucleus on the agents position.

Theorem 3.3.1 *If the agents are in the critical points of the function $\mathcal{G}_t$, then with respect to the defined teams nucleus l_t in (3.6), it is concluded that the function $\mathcal{G}$ is minimized.*

Proof 2 *According to (3.14), the function $\mathcal{G}_t$ is minimized when $p_{tm} = C_{V_{tm}}$. Assuming that all the agents inside the teams are in the centroid of their respective Voronoi cells, the nucleus of the teams can be found as*

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} C_{V_{tm}}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{3.15}$$

Substituting the centroid of the agents Voronoi in (3.4), we have

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} \int_{V_{tm}} q\ \rho(q) dq / M_{V_{tm}}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{3.16}$$

Since the density distribution function is positive on $\mathcal{Q}$, each Voronoi cell has a nonzero measure, i.e., $M_{V_{tm}} \neq 0$ for $m = \{1, 2, \ldots, n_t\}$. Hence, we obtain

$$l_t = \frac{\sum_{m=1}^{n_t} \int_{V_{tm}} q\ \rho(q) dq}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{3.17}$$

Considering that the Voronoi cells share no interior space with each other and also due to the continuity of the density function ρ over $\mathcal{Q}$, it can be concluded that

$$l_t = \frac{\sum_{m=1}^{n_t} \int_{V_{tm}} q\ \rho(q) dq}{M_{V_t}} = \frac{\int_{V_t} q\ \rho(q) dq}{M_{V_t}} = C_{V_t}. \tag{3.18}$$

Therefore, it was shown that when agents are in their respective centroids, the nucleus of team also is in the centroid of the team's Voronoi cell.

3.3.2 TEAM-BASED LLOYD ALGORITHM

To minimize the cost functions $\mathcal{G}$ and $\mathcal{G}_t$ over time, an extension of Lloyd algorithm [32, 37] is proposed here. The proposed algorithm should consider updating the team Voronoi cells while evolving the positions and partitions of the agents inside the teams. According to Theorem 3.3.1, if the agents move towards the centroid of their Voronoi cells, the nucleus of each team also evolves towards its associated centroid. Hence, the two level optimization problem can be seen as the problem of assigning agents to their associated regions inside a Voronoi cell with changing boundaries. The following dynamics is enforced on the agents

$$\dot{p}_{tm} = u_{tm} \tag{3.19}$$

Considering the cost function $\mathcal{G}_t$ as a Lyapunov function guarantees the stability of the agents by moving them to their associated local minima [32]. This results in the following control law

$$u_{tm} = -k(p_{tm} - C_{V_{tm}}), \tag{3.20}$$

where k is a positive gain and the Voronoi cells are being updated continuously over time.

Corollary 3.3.1.1 *For the closed-loop system obtained by controller (3.20), the sets of centroidal Voronoi configurations are subspaces of $\mathcal{Q}$ and $\mathcal{Q}_t$. If the set of centroidal Voronoi configurations on $\mathcal{Q}$ is finite, the teams nucleus and agents location converge to centroidal Voronoi configurations.*

Proof 3 *By applying the control law (3.20), the agent positions converge asymptotically to the set of critical points of $\mathcal{G}_t$ according to Proposition 3.1 in [32], $p_{tm} \rightarrow C_{V_{tm}}$. It is concluded that $l_t \rightarrow C_{V_t}$ by considering $p_{tm} \rightarrow C_{V_{tm}}$ in equation (3.6) with a similar argument as in the proof of Theorem 3.3.1. Therefore, the teams nucleus and agents locations converge to the sets of centroidal Voronoi configurations. If the set of centroidal Voronoi configurations on $\mathcal{Q}$ is finite, the limit of $\mathcal{P}_t$ is unique and equals to one of the centroidal Voronoi configurations. Consequently, the teams nucleus converges to a centroidal Voronoi configuration according to Theorem 3.3.1.*

Remark 3.1 *The region assigned to each team changes over time. Since the assigned region is time-varying, there may exist robots which are located outside the assigned region of their team. Consequently, there might be empty Voronoi cells which belonged to the outside agents. In this case, the outside agents move towards to the centroids of their Voronoi cells by using the controller (3.20).*

Remark 3.2 *According to Remark 3.1, there is a possibility that an agent can be located in a region covered by another team. While this agent is moving toward the assigned region of its team, it may collide with agents of another team. We consider the coverage problem without the possibility of agents colliding with each other.*

3.4 Modeling A Distributed Network of Agents

The agents need to be modeled with respect to their actions including sensing, communication, computation and control. The behavior of the agents in a given network are then describable as the interaction of the agents to perform the assigned task. In essence, the communication network and the data flow in the system of agents based on the proposed algorithms need to be investigated. The agents communication characteristics and protocols are discussed in this section.

3.4.1 CHARACTERISTICS OF THE AGENTS COMMUNICATION

An agent in the given framework is introduced here as the m^{th} element of the t^{th} team in a given network. Each agent is equipped with processor that is capable of allocating the state of the system and performing the required operations. For the agent that is located at p_{tm} and can move over the space at any time for any period of time $\delta t_{tm} \in \mathbb{R}_+$ following the enforced first order dynamics (3.19). The processor has access to both agent position p_{tm} and its associated team nucleus l_t that is received through the communication with the neighboring agents. The processor also computes the control pair $(\delta t_{tm}, u_{tm})$ for the robot at p_{tm}. The agents on the borders of teams also receive the information for the nucleus of the neighboring teams. This information is provided by communicating with the neighboring agents belonging to the neighboring teams. Furthermore, it can detect the neighboring agents within the radius R_{tm} of the agent at p_{tm}. In addition to this, the processor is able to send/receive information to/from the other agents within the communication radius $R_{tm} \in \mathbb{R}_+$. It is also assumed that the processor can adjust the radius R_{tm} regarding the minimum required communication radius in a limited communication bandwidth.

3.4.2 TEAM AND AGENT'S VORONOI MAINTENANCE

In order to calculate the Voronoi partition, each agent needs to know the position of the other agents in its neighborhood and also the agents on the borders require the nucleus of the neighboring teams to calculate the borders of the teams. To this end, the designed communication network should provide the motion control scheme with the required information. In this section, the Voronoi computation and maintenance is handled by the proposed algorithm for the asynchronous communication to maintain the data flow inside each team and among the teams.

To initialize, it is assumed that all the agents have the nucleus of their associated team and other agents position at the initial time. The agents employ an adjustable communication radius based algorithm as presented in Algorithm 1 to receive the position information for computing the Voronoi cells. It should be mentioned that since, the agents on the boundaries need the nuclei of the other teams to compute their Voronoi cells, Algorithm 1 guarantees a proper communication radius

to receive the information from the neighboring teams while making sure that the interior agents will not be increasing their communication radius unnecessarily. To this aim, two communication radiuses are defined, first, the interior agents communicate by an adjustable radius R_{c1} to receive the data from the neighboring agents belonging to the same team. Then, if an agent detects the agents belonging to the neighboring teams, it means the associated agent is on the boundaries, therefore, the communication radius is adjusted to R_{c2} to gather the nucleus of the neighboring teams. Finally, when the two Voronoi cells are obtained by (3.2) and (3.3) then the intersection of the two cells is computed as the final Voronoi cell for the agents on the boundaries.

Due to the nature of the team based communication and considering the fact that agents need to know the nucleus of the team, the communication network is designed to transfer data throughout each team's network towards the nucleus. Then, the updated nucleus is distributed among the agents of each team.

To maintain the Voronoi cells, the nucleus of the team needs to be updated. To do so, we propose Algorithm 2 based on the greedy forwarding strategy presented in [28] that sends the data pocket that contains information regarding mass of the Voronoi cells and the positions of the robots towards the nucleus of the team. The information pocket that is sent is consisted of two data strings, the first one contains the summation over the multiplication of the mass and the location of each agent and the second string represents the summation over the multiplication of the mass of the Voronoi cells that the data pocket has been passed through. In other words, the algorithm is designed in a way that each agent sends the information to the agent closest to the nucleus in its neighborhood.

Every agent that receives the data pocket sums up all the received data and forwards it towards nucleus.

Every node in the network performs the forwarding process until the time based stopping criteria is satisfied that is defined by multiplying the number of agents in each team by the maximum operating period of time $\max\{\delta t_{t1}, \ldots, \delta t_{tn_t}\}$ for t^{th} team. The agent closest to the nucleus performs sums up all the received data and calculates the new nucleus of the team.

Algorithm 1 Sensing algorithm for teams of robots, inspired from [32]

initialize sensing radius R_{tm}
detect all agents p_{tm} in the same team and all nucleus l_t of the neighboring team's agents within the radius R_{tm}
update $P^m(t_m)$ and $\mathcal{L}^m(t_m)$ to compute $\mathcal{Q}(p_{tm}, R_{tm})$ and $\mathcal{Q}(l_t^m, R_{tm})$
$R_{c1} = \max_{q \in \mathcal{Q}(p_{tm}, R_{tm})} \|p_{tm} - q\|$
if detect neighboring team's agents **then**
    $R_{c2} = \max_{q \in \mathcal{Q}(l_t^m, R_{tm})} \|l_t^m - q\|$
else
    $R_{c2} = 0$
end if
while $R_{tm} \leq 2\max\{R_{c1}, R_{c2}\}$ **do**
    $R_{tm} = 2\max\{R_{c1}, R_{c2}\}$
    detect all agents p_{tm} in the same team and all nucleus l_t of the neighboring team's agents within the radius R_{tm}
    update $P^m(t_m)$ and $\mathcal{L}^m(t_m)$
    compute $\mathcal{Q}(p_{tm}, R_{tm})$ and $\mathcal{Q}(l_t^m, R_{tm})$
    $R_{c1} = \max_{q \in \mathcal{Q}(p_{tm}, R_{tm})} \|p_{tm} - q\|$
    if detect neighboring team's agents **then**
        $R_{c2} = \max_{q \in \mathcal{Q}(l_t^m, R_{tm})} \|l_t^m - q\|$
    else
        $R_{c2} = 0$
    end if
end while
$R_{tm} = 2\max\{R_{c1}, R_{c2}\}$
$V_{tm} = \mathcal{Q}(l_t^m, R_{tm}) \cap \mathcal{Q}(p_{tm}, R_{tm})$

As soon as the agent closest to the nucleus calculates the updated nucleus, it disseminates the data pocket over the team network by the communication radius that has been already calculated in the Voronoi computation. Each agent that receives the new nucleus also runs the same data transfer algorithm until the data pocket reaches to the borders. To realize whether agents are sharing boundaries with other teams, each agent can run a quick check to see if any of its boundaries has been drawn by using the teams nucleus or not. This is considered as the stopping criterion for the message routing. Now, the agents have been informed about the new nucleus of the team, and hence, the ones on the border communicate with the agents belonging to the other teams that they share

Algorithm 2 The greedy forwarding algorithm for nucleus calculation inspired by [28]

Require: given $\delta t_{t1}, \ldots, \delta t_{tn_t}$

agent m detects agent m' as the neighbor closed agent towards the nucleus l_t
if $m' \neq m$ **then**
    send the pair $(M_{V_{tm}} p_{tm}, M_{V_{tm}})$ to the agent m'
else$(\alpha_t, \beta_t) = (M_{V_{tm}} p_{tm}, M_{V_{tm}})$
end if
$T_m = \delta t_{tm}$
while $T_m < n_t \max\{\delta t_{t1}, \ldots, \delta t_{tn_t}\}$ **do**
    $T_m \leftarrow T_m + \delta t_{tm}$
    if receives data pocket **then**
        sum up all the received data
        if $m' \neq m$ **then**
            send the summation to the agent m'
        end if
        if $m' = m$ **then**
            $(\alpha_t, \beta_t) \leftarrow (\alpha_t, \beta_t)+$ the summation of all the received data pockets
        end if
    end if
end while
if $m' = m$ **then**
    compute the update nucleus by (3.6) as $l_t = \alpha_t / \beta_t$.
end if

boundaries with to get their team nucleus to draw their Voronoi cells. The algorithm for drawing the associated Voronoi regions of the agents that might require communication with other teams is given in Algorithm 1. Now that the team boundary has been updated, the agents need to repartition their associated space.

3.5 SIMULATION AND RESULTS

In this section, the presented team based partitioning is illustrated through two simulation examples. 50 agents are deployed on a 5×10 rectangular in five teams with equal number of agents. To illustrate the case that there are multiple important regions, two Gaussian functions with 3 and 5 picks are chosen respectively. The density function is chosen as a Gaussian function represented

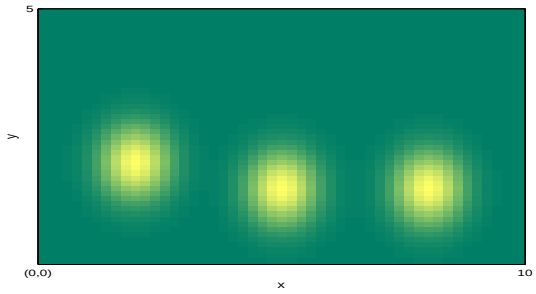


Figure 3.1: Gaussian density function with chosen centers as $(2, 2), (5, 1.5)$ and $(8, 1.5)$.

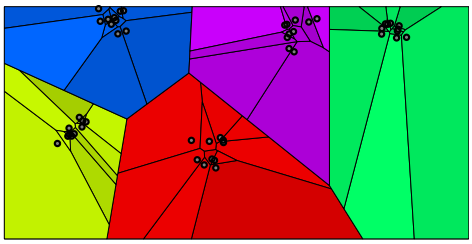
by

$$\varphi(x) = \sum_{i=1}^{n_p} \exp(-\frac{(x - a_i)^2 + (y - b_i)^2}{2\sigma^2}), \tag{3.21}$$

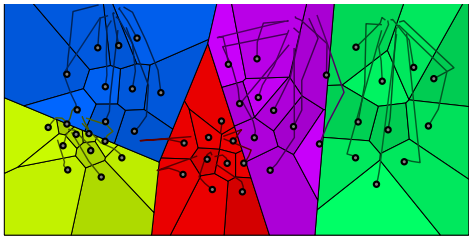
where n_p is the number of centers, (a_i, b_i) represents the coordinate of the centers and σ is the variance.

EXAMPLE 1

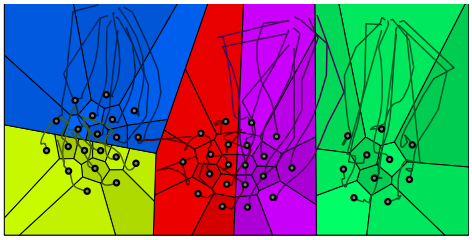
The first density function is chosen as a Gaussian function given by (3.21) where $n_p = 3$ and $\sigma = 0.25$. Figure 3.1 illustrates the density function over the given space. A random configuration is chosen as the initial configuration of the nucleus of the teams. Then, the robots are deployed in the region associated with each team. As the proposed control law moves the agents to the centroid of their Voronoi cells, the teams nucleus also moves towards their centroid. Figure 3.2 shows the solution to the coverage problem. It can be seen that along with partitioning in the team level the agents also divide the region assigned for each team considering the given density function. Also, regarding the interdependency of teams and their interior agents, the teams spread over the given space to provide the optimum coverage while their interior agents also partition each team Voronoi cell into smaller partitions considering the main density function. The trend shows that the teams also tend to concentrate on the region given by the density function, hence, the team partitioning divides the space among the teams collaboratively. As it can be seen in Figure 3.4c, the teams on



(a)



(b)



(c)

Figure 3.2: The change from the initial to the final configuration is shown from (a) to (c) for example 1.

the left and in the middle divide the region in a way that they optimize their associated sensing function according to the defined cost function both in the team level (3.7) and agents level (3.8) for the best coverage. The traversed path by each agent to reach to the optimum configuration is shown in Figure 3.4c.

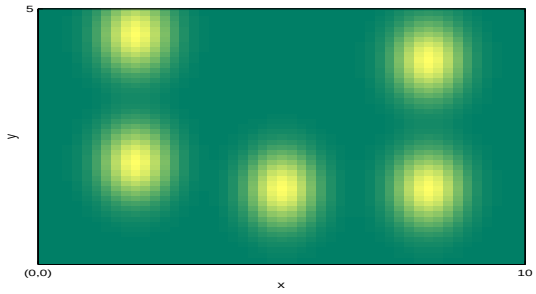


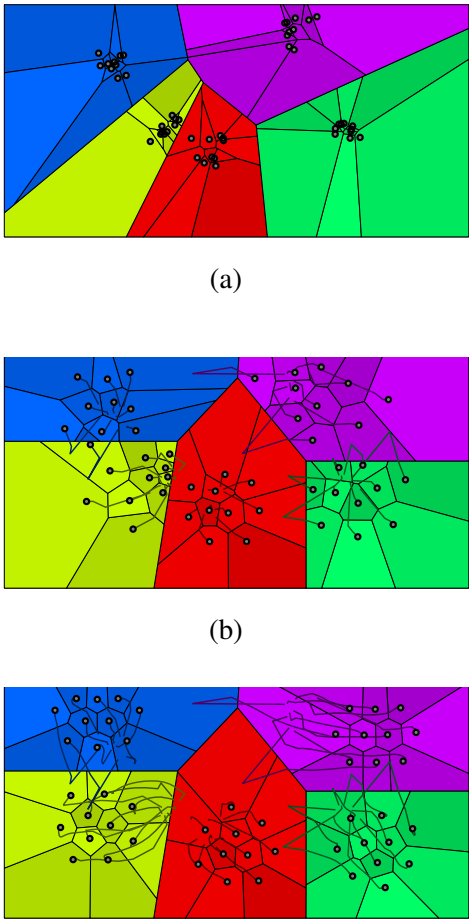
Figure 3.3: Gaussian density function with chosen centers as $(2, 2), (5, 1.5)$ and $(8, 1.5), (8, 4), (2, 4.5)$.

EXAMPLE 2

As the second example, a different density function is chosen to study the presented team based coverage when there are higher number of equally important regions. The Gaussian function is chosen as given in (3.21) where $n_p = 5$ and $\sigma = 0.25$. Figure 3.3 illustrates the Gaussian function with 5 centers that represent the regions with high importance in the coverage context. The results for the coverage problem is shown in Figure 3.4 where the initial, transient and final states of the agents are illustrated. The simulation results indicate that the proposed coverage approach can cope with complexity of the problem arising from the higher number of the significant regions. This can represent the flexibility of the team based coverage to deal with multiple assigned tasks.

3.6 CONCLUDING REMARKS

To adapt with the complexity of the coverage problems in real life applications, a team-based coverage approach is presented. The proposed approach solves the problem in a more local way meaning that the main region is divided into multiple subregions associated with multiple deployed teams to handle the tasks that require a more diverse group of robots regarding its complexity. This is to say that due to the diversity of the coverage problems the need for deploying robots with different dynamics or communication characteristics might be inevitable, hence, the team based



(a)

(b)

(c)

Figure 3.4: The change from the initial to the final configuration is shown from (a) to (c) for example 2.

approach is an attempt to modify present methods in a way that the teams of robots can divide the region among themselves based their capabilities. This provides the means for autonomous deployment of multiple teams of robots when there is need for different types of robots form both communication and dynamics perspective.

CHAPTER 4

A TEAM-BASED APPROACH FOR COVERAGE CONTROL OF MOVING SENSOR NETWORKS [1]

[1] F. Abbasi, A. Mesbahi and J. Mohammadpour: A Team-based Approach for Coverage Control of Moving Sensor Networks. 2016. *Submitted to Automatica, Under the Second Round of Revision.*

ABSTRACT

In this paper, the coverage control problem is examined for mobile sensor networks from a *team*-based perspective. The objective is to deploy a group of robotic teams aiming at minimizing the serving cost based on a given density function that defines the probability of events in the environment. A team-based approach is introduced here to formulate a coverage scheme for the case where, for a variety of reasons, e.g., heterogeneity in their embedded communication capabilities or the dynamics, the robots are required to be kept in the same team with other similar agents. This also allows for adjusting to the changes in the environment or the assigned task(s) in a collaborative way as a team. The main cost function is defined as the total sensing cost for the agents that form the teams. The immediate consequence of this formulation is that dynamics on the boundary of the teams must be taken into account for the agents that share boundaries with their neighboring teams. A gradient decent-based control law is developed to ensure the optimal deployment of the agents within each team while making sure that the team dynamics also take part in the final optimal solution. To further elaborate on the use of the presented formulation, a formation problem is defined for the teams of agents with either the same or different formation structures. It is shown that the optimal desired formation comes with the cost of sacrificing the sensing performance. Finally, numerical results are presented to illustrate the effectiveness of the team-based partitioning and formation methods that enable the distributed deployment of the heterogeneous teams of agents.

4.1 INTRODUCTION

There have been advancements in developing techniques for deploying a group of robots in a given environment to perform assigned distributed tasks. Examples of such tasks include surveillance, search and rescue operations, sensing, and data collection [10, 78, 85, 151]. The core problem can be seen as a workload sharing task to allocate the share of each agent from the total workload. The underlying optimization problem is NP-hard (Non-deterministic Polynomial-time hard) [46], and hence, finding a local minimizer is desired. For instance, assuming that the agents are identical, the

partitioning task is carried out to divide the area of interest among the agents equally in space and time, which leads to an optimal partitioning in case of both deterministic [14, 101] and uncertain data [97]. Among different feasible types of partitioning, the Voronoi cells offer the optimal solution for the space sharing tasks [32]. Depending on the way the partitioning is defined, there are various kinds of Voronoi cells such as power diagrams, additively-weighted and multiplicatively-weighted Voronoi partitions as introduced in [11, 126].

The existing approaches for the coverage control are based on the assumption that all agents belong to a single team [109]. However, this assumption is not realistic in many real-world applications, as the agents may differ from , e.g., dynamics or communication perspective [135]. A multi-robot system can generally be considered as a homogeneous or heterogeneous system depending on the similarities or differences in their properties, e.g., desired performance index, dynamics, etc., that is required when coping with various complex assigned tasks [65, 134, 155]. In the present work, a new coverage strategy is proposed that aims at taking into account the differences in the robots dynamics by offering a team-based design approach, where, each robot might team up with others based on its assigned task, associated dynamics or embedded communication capabilities. This would make it possible to improve reliability and flexibility of the deployment algorithm. It should be noted that throughout this work, it is assumed that the structure of the teams and the agents within each team is known a priori.

The present work introduces a new team-based coverage control scheme which can handle different scenarios in heterogeneous systems of robots. The proposed approach addresses the problem of agents deployment by considering teams of robots instead of evaluating each agent individually. The main problem can be defined as a general optimization problem over all the teams and their associated agents, where the cost function for each team is defined over the Voronoi of each team with dynamic boundaries. The agents will move towards their local minima until the optimal configuration is achieved. In the proposed approach, firstly, a local minimum to the deployment problem is obtained within each team to generate the Voronoi cells of the agents inside the team. Then, the nucleus of the team is calculated through the defined weighted mean formula that gen-

erates the Voronoi cell associated with each team. By considering the dependency of the team nucleus on the position of agents in that team, it is proven that both agents and their respective team's nucleus essentially converge to their optimum configuration.

The agents in each team can be classified into two groups of interior members and members on the boundaries based on whether or not they share a boundary with the agents belonging to other teams. Each interior member is able to compute its own Voronoi cell by only knowing the location of its neighbors. The location of the neighbors is obtained through a modified adjust-communication-radius algorithm which has been presented in [32]. However, this does not hold true for the members on the boundaries. These members need not only the location of the neighbor robots in their associated team, but also the nuclei of the neighboring teams with their team Voronoi cells. The agents on the boundaries can obtain the nuclei of neighboring teams through the use of the communication algorithm proposed in [3].

As an application of the proposed team-based method, we study formation control problem. The formation control methods can be generally divided into three categories: behavior-based, virtual structure formation, and leader-follower formation [89, 100, 139]. In the latter category of leader-follower formation, one agent is designated as the leader and the others as followers [62]. The follower agents essentially follow the leader agent with a desired distance and separation bearing angle. The formation of agents around a certain virtual point considered as the virtual leader has been studied in nature-inspired flocking schemes under the leader-follower framework in [103, 136]. To ensure certain dynamic behaviors such as moving with the same speed as the leader, the agents take a certain formation and stay in a relatively close distance to their nearby agents. In this paper, the presented team-based scheme is shown to offer the ability to impose certain formations on the agents belonging to each team while pursuing the main coverage task. The basic idea is to adjust the distance of agents to the nucleus of their associated team by introducing an additional formation term in the main coverage sensing function. The additional term ensures a certain distance from the nucleus by changing the formation factor. This can be viewed as a factor

that enforces the agent to either expand or compress with respect to the desired formation. Different formations can be also achieved through selection of various formation factors.

The contributions of the present work are threefold. First, incorporating the concept of team into the sensing coverage through defining a team-based sensing performance function provides a flexible formulation for coverage related tasks to deploy heterogenous teams of robots and assign various tasks. Second, the proposed team-based approach enables forming various teams consisting of (possibly) different number of agents within the main coverage framework that can be translated to deploying more robots to the regions with higher degree of importance. Finally, by giving a group of robots a team entity, they can maintain certain configurations due to a number of constrains or objectives, e.g., the limited communication radius or acquiring a better coverage by maintaining a certain formation while ensuring an optimal coverage through collaborating with other teams.

The remainder of this paper is structured as follows. Definitions and the problem statement are provided for the team-based coverage control in Section II. Section III introduces an approach for formation control of teams of agents. Section IV presents numerical simulation results to illustrate the team-based partitioning and proposed formation control.

4.1.1 NOTATIONS

We use $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$ to respectively denote the sets of natural, real, and nonnegative real numbers. Throughout the paper, I_r denotes $r \times r$ identity matrix. We define $\mathcal{Q}$ as a convex polytope in $\mathbb{R}^2$ and let $\mathscr{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_t\}$ be a *partition* of $\mathcal{Q}$ as a collection of t closed subsets with disjoint interiors. The boundary of $\mathcal{Q}$ is denoted by $\partial\mathcal{Q}$. Moreover, the so-called *distribution density function* is denoted by φ where $\varphi : \mathcal{Q} \to \mathbb{R}_+$ represents the probability of some phenomenon occurring over space $\mathcal{Q}$. The function φ is assumed to be measurable and absolutely continuous. The Euclidean distance function is denoted by $\|\cdot\|$, and $|\mathcal{Q}|$ represents the Lebesgue measure of convex subset $\mathcal{Q}$. The vector set $\mathcal{P}_t = (p_{t1}, p_{t2}, \ldots, p_{tn_t})$ is the location of n_t agents belonging to t^{th} team.

4.2 A TEAM-BASED APPROACH FOR COVERAGE CONTROL

In the coverage control literature, the *locational cost function* is defined as a measure of the sensing performance. In this work, we present a modified version of the locational function that is suitable for the proposed team-based method. The function is defined as

$$\mathcal{H}(P, \mathcal{Q}) = \sum_{i=1}^{N} \int_{\mathcal{W}_i} f(\|q - p_i\|)\varphi(q)\mathrm{d}q, \tag{4.1}$$

where for n teams, $N = \sum_{t=1}^{n} n_t$, n_t is the number of agents belonging to the t^{th} team, and P is the set of all agents. Assuming that i^{th} agent is assigned to the region $\mathcal{W}_i$, the objective is to minimize the cost function $\mathcal{H}$ by finding the optimal locations of the agents and their assigned regions $\mathcal{W}_i$ whose union is $\mathcal{Q}$. The sensing performance of the agents should decay as we move away from their location, and hence, sensing performance can be evaluated as a function of distance from the agent, i.e., $f(\|q - p_i\|)$, where $q \in \mathcal{Q}$. In this context, all the agents are assigned over the space no matter how the agents can collaborate or coordinate with each other locally. The team-based partitioning of the agents introduced in this paper addresses this by dividing agents into multiple teams pursuing assigned tasks.

4.2.1 VORONOI PARTITIONS

The main objective of this work is to adopt a team-based concept in the agents deployment and partitioning framework. To achieve this, we first need to define an optimization problem that can handle not only the deployment and partitioning tasks inside teams but also the partitioning inside the defined polytope $\mathcal{Q}$. This optimization problem should consider the agents individual cost function, as well as their accumulated cost within their teams. To start with, we define the set of teams by $\mathcal{L} = (l_1, l_2, \ldots, l_n)$ where $l_t, t = 1 : n$, represents the nucleus of the t^{th} team team t that is a function of the agents position in the associated team, i.e., $l_t = g(p_{t1}, p_{t2}, \ldots, p_{tn_t})$. The dependency of l_t on the position of the agents is discussed later. Next, we partition the polytope $\mathcal{Q}$ into a set of Voronoi cells $\mathcal{V}(\mathcal{L}) = \{V_1, V_2, \ldots, V_n\}$ considered as the optimal partitioning for a set of

agents with fixed locations at a given area as

$$V_t = \{q \in \mathcal{Q} |\ \|q - l_t\| \leq \|q - l_s\|, s = 1, \ldots, n; s \neq t\}. \tag{4.2}$$

The obtained Voronoi cells associated with the nuclei of the teams are then considered as the convex polytope set to deploy their associated agents. Therefore, the sub-partitions are defined on the basis of the Voronoi cells V_t obtained from the team level partitioning. The Voronoi partitions $\mathcal{V}_t(\mathcal{P}_t) = \{V_{t1}, V_{t2}, \ldots, V_{tn_t}\}$ generated by the agents $(p_{t1}, p_{t2}, \ldots, p_{tn_t})$ belonging to the t^{th} team are defined as

$$V_{tm} = \{q \in V_t |\ \|q - p_{tm}\| \leq \|q - p_{tr}\|, r = 1, \ldots, n_t, r \neq m\}, \tag{4.3}$$

where p_{tm} denotes the location of m^{th} agent in t^{th} team for $m \in \{1, \ldots, n_t\}$.

The agents in each team are divided into two subgroups, boundary and interior groups, where the cells associated with each group require a different set of data, i.e., their neighbors' position, to be maintained. The interior group represents the agents that share boundaries only with the agents belonging to the same team while the agents in the boundary group have neighbors not only in the same team but also in the neighboring teams-they may also share boundaries with the convex polytope $\mathcal{Q}$. In general, the boundary associated with each agent ∂V_{tm} is either an edge shared with the agents within the same team or edges shared with the teams in the neighborhood depending on the position of the agent within the team. The agents in the boundary group share at least one edge with other teams. An edge that is shared with the neighboring agent f in the same team is shown by $\partial V_{tm,f}$. The edges associated with the agents in the boundary group that are shared with the neighboring team k and main convex polytope $\mathcal{Q}$ are represented by ∂V_{tr}^k and ∂V_{tr}^0, respectively. Figure 4.1 illustrates the boundaries and their normal vectors for Voronoi V_{tm}. It is noted that the agents in the boundary group may share boundaries with the agents in the interior group where the same notation as the boundaries of the interior agents is used to represent these edges.

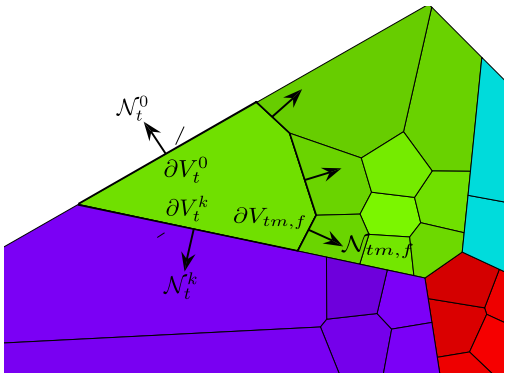


Figure 4.1: An illustrative example of team-based Voronoi partitioning, where the boundaries and their associated normal vectors for the Voronoi cell V_{tm} are shown for the edges shared with both team k in its neighborhood and agent tf, i.e., f^{th} agent in the t^{th} team, in the same team in addition to the edge ∂V_t^0 shared with the main region $\mathcal{Q}$.

We recall the basic characteristics of the Voronoi partitions including their associated mass, centroid, and polar moment of inertia defined as [32]

$$\begin{aligned} M_{V_{tm}} = \int_{V_{tm}} \varphi(q)\mathrm{d}q,\ C_{V_{tm}} = \frac{1}{M_{V_{tm}}} \int_{V_{tm}} q\ \varphi(q)\mathrm{d}q, \\ J_{V_{tm},p_{tm}} = \int_{V_{tm}} \|q - p_{tm}\|^2 \varphi(q)\mathrm{d}q. \end{aligned} \tag{4.4}$$

In addition to the parameters defined for the Voronoi cells of each agent inside the teams, we also need to define the characteristics of the teams's Voronoi cells. According to the definition of the Voronoi partitions, it can be deduced that

$$M_{V_t} = \sum_{m=1}^{n_t} M_{V_{tm}},\ C_{V_t} = \frac{1}{M_{V_t}} \int_{V_t} q\ \varphi(q)\mathrm{d}q. \tag{4.5}$$

The nucleus of the team that is a function of the agents position is defined as

$$l_t = \frac{\sum_{m=1}^{n_t} M_{V_{tm}} p_{tm}}{\sum_{m=1}^{n_t} M_{V_{tm}}}. \tag{4.6}$$

As described earlier, the nucleus is a representative of the agents position in the team and can be considered as the collective position of the agents that is needed for the computation of the Voronoi diagram of the teams, i.e., V_t.

4.2.2 FORMULATION OF THE LOCAL OPTIMIZATION PROBLEM

The deployment task in the presented team-based framework can be addressed by solving an optimization problem over the convex polytope $\mathcal{Q}$ where each team's region needs to be optimized in the sense of its associated sensing function. Hence, the following cost function is considered

$$\mathcal{G}(\mathcal{L}, \mathscr{Q}) = \sum_{t=1}^{n} \mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t), \tag{4.7}$$

where $\mathcal{G}_t$ is the cost function associated with the sensing performance of the agents belonging to the t^{th} team. The sensing performance is considered as $f(\|q-p_{tm}\|) = \|q-p_{tm}\|^2$ for the m^{th} agent belonging to the t^{th} team. The solution to (4.7) gives a local minimum to the deployment problem where agents are considered as the members of various collaborating teams. We define a set of polygons as $\mathscr{Q}_t = \{\mathcal{Q}_{t1}, \mathcal{Q}_{t2}, \ldots, \mathcal{Q}_{tn_t}\}$, with disjoint interiors, whose union is V_t. The following cost function is defined for each team as a function of the position of its associated agents

$$\mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t) = \sum_{m=1}^{n_t} \int_{\mathcal{Q}_{tm}} \|q - p_{tm}\|^2 \varphi(q) \mathrm{d}q. \tag{4.8}$$

Remark 4.1 *It is proven that among different partitioning schemes, the Voronoi partitions are optimum (for a single team) in the sense of minimizing the defined cost function (4.8) [32]. Hence, for a given set of agents with position $\mathcal{P}_t \in V_t$ and a partition $\mathscr{Q}_t$ of V_t, we have*

$$\mathcal{G}_t(\mathcal{P}_t, \mathcal{V}_t(\mathcal{P}_t)) \leq \mathcal{G}_t(\mathcal{P}_t, \mathscr{Q}_t), \tag{4.9}$$

which implies that the Voronoi cells represent the optimum partitioning of the area associated with each team.

The next step is to obtain (locally optimum) location of the agents and the nucleus of their associated team. The derivative of the cost function (4.7) associated with n teams, each of which consists of n_t agents is

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = \frac{\partial}{\partial p_{sm}} \sum_{t=1}^{n} \sum_{r=1}^{n_t} \int_{V_{tr}} \|q - p_{tr}\|^2 \varphi(q) \mathrm{d}q, \quad s = 1, \ldots, n, \quad m = 1, \ldots, n_t. \tag{4.10}$$

The solution to this optimization problem differs from that to the conventional sensing cost functions due to the previously defined dependency of the boundaries of the boundary agents on their

associated nuclei that are functions of agents position. The derivative with respect to the coordinates of agent in p_{sm} is obtained as

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = \int_{V_{sm}} \frac{\partial}{\partial p_{sm}} \|q - p_{sm}\|^2 \varphi(q) \mathrm{d}q + \sum_{t=1}^{n} \Bigg(\sum_{r=1}^{n_{it}} \int_{\partial V_{tr}} \|q - p_{tr}\|^2 \varphi(q) \frac{\partial \partial V_{tr}}{\partial p_{sm}} \mathcal{N}_{tr} \mathrm{d}q + \sum_{r=1}^{n_{bt}} \int_{\partial V_{tr}} \|q - p_{tr}\|^2 \varphi(q) \frac{\partial \partial V_{tr}}{\partial p_{sm}} \mathcal{N}_{tr} \mathrm{d}q \Bigg), \quad (4.11)$$

where n_{it} is the number of the interior agents, and n_{bt} is the number of agents in the boundary group in the t^{th} team. It can be inferred from the team-based partitioning that the boundaries of the Voronoi cell V_{tr} are either directly or indirectly dependent on p_{tm}. The direct dependency is obviously resulted from the definition of the interior Voronoi cells (4.3). We note that V_{tm} and the voronoi cells in its neighborhood are directly dependent on p_{tm}. The indirect dependency can be seen in the boundary agents, where they share at least one edge with the agents in the neighboring teams (or contribute at least one edge to the boundary of the team).

Remark 4.2 *The boundaries shared with other teams are indirectly dependent on p_{tm} via the definition of the nucleus of the team l_t. Also, the Voronoi cells of the interior agents in other teams are independent of p_{tm} leading to $\frac{\partial \partial V_{tr}}{\partial p_{sm}} = 0$ for $t \neq s$.*

Hence, the term associated with the interior agents in (4.11) can be reduced to the integral on the boundaries of the agent p_{tm} and the ones that it shares boundaries with as follows

$$\sum_{r=1}^{n_{is}} \int_{\partial V_{sr}} \|q - p_{sr}\|^2 \varphi(q) \frac{\partial \partial V_{sr}}{\partial p_{sm}} \mathcal{N}_{sr} \mathrm{d}q = \sum_{f \in \mathcal{F}} \int_{\partial V_{sm,f}} \|q - p_{sm}\|^2 \varphi(q) \frac{\partial \partial V_{sm,f}}{\partial p_{sm}} \mathcal{N}_{sm,f} \mathrm{d}q + \sum_{f \in \mathcal{F}} \int_{\partial V_{sf,m}} \|q - p_{sr}\|^2 \varphi(q) \frac{\partial \partial V_{sf,m}}{\partial p_{sm}} \mathcal{N}_{sf,m} \mathrm{d}q, \quad (4.12)$$

where $\mathcal{N}_{sf,m}$ is the normal vector associated with the edge $\partial V_{sf,m}$, and $\mathcal{F} = \{f | p_{sf} \in \mathcal{N}_{p_{sm}}\}$ with $\mathcal{N}_{p_{sm}}$ representing the set of agents that share boundaries with agent p_{sm}.

Remark 4.3 *The integral on each boundary shared with the neighboring agents is identical for agents on both sides expect that the normals will have opposite signs, i.e., $\mathcal{N}_{sm,f} = -\mathcal{N}_{sf,m}$.*

Hence, we have

$$\sum_{f\in\mathcal{F}}\int_{\partial V_{sm,f}}\|q-p_{sm}\|^2\varphi(q)\frac{\partial\partial V_{sm,f}}{\partial p_{sm}}\mathcal{N}_{sm,f}dq=-\sum_{f\in\mathcal{F}}\int_{\partial V_{sf,m}}\|q-p_{sr}\|^2\varphi(q)\frac{\partial\partial V_{sf,m}}{\partial p_{sm}}\mathcal{N}_{sf,m}dq. \tag{4.13}$$

Using Remark 4.3, the second term in (4.11) is equal to zero. Moreover, the boundaries shared with $\partial\mathcal{Q}$ have no dynamics which results in $\frac{\partial\partial V_{tr}^0}{\partial p_{sm}}=0$. Therefore, considering Remark 4.2, the third term in (4.11) can be written as

$$\begin{aligned}\sum_{t=1}^{n}\sum_{r=1}^{n_{bt}}\int_{\partial V_{tr}}\|q-p_{tr}\|^2\varphi(q)\frac{\partial\partial V_{tr}}{\partial p_{sm}}\mathcal{N}_{tr}\mathrm{d}q=&\sum_{f\in\mathcal{F}}\int_{\partial V_{sm,f}}\|q-p_{sm}\|^2\varphi(q)\frac{\partial\partial V_{sm,f}}{\partial p_{sm}}\mathcal{N}_{sm,f}\mathrm{d}q+\\&\sum_{f\in\mathcal{F}}\int_{\partial V_{sf,m}}\|q-p_{sm}\|^2\varphi(q)\frac{\partial\partial V_{sf,m}}{\partial p_{sm}}\mathcal{N}_{sf,m}\mathrm{d}q+\\&\sum_{t\in\{s,g|l_g\in\mathcal{N}_{l_s}\}}\sum_{r=1,k\in\mathcal{K}}^{n_{bt}}\int_{\partial V_{tr}^k}\|q-p_{tr}\|^2\varphi(q)\frac{\partial\partial V_{tr}^k}{\partial p_{sm}}\mathcal{N}_{tr}^k\mathrm{d}q,\end{aligned} \tag{4.14}$$

where $\mathcal{K}=\{0,g|l_g\in\mathcal{N}_{l_t}\}$ and $\mathcal{N}_{l_t}$ is the set of teams neighboring with team t. Also, $\mathcal{N}_{tr}^k$ in (4.14) denotes normal vector associated with the edge ∂V_{tr}^k. The first two terms in the right hand side of (4.14) represent the integral on the shared boundaries of the V_{sm} with the agents belonging to the same team. Hence, it can be concluded from Remark 4.3 that these terms cancel each other out. Moreover, the boundaries shared with $\partial\mathcal{Q}$ have no dynamics that results in $\frac{\partial\partial V_{tr}^0}{\partial p_{sm}}=0$. Due to the dependency of the team boundaries on the nucleus, only the boundaries of the agents in the boundary group shared with their associated team should be considered for the integration. Substituting (4.13) and (4.14) into (4.11) results in

$$\frac{\partial\mathcal{G}}{\partial p_{sm}}=\int_{V_{sm}}\frac{\partial}{\partial p_{sm}}\|q-p_{sm}\|^2\varphi(q)\mathrm{d}q+\sum_{t\in\{s,g|l_g\in\mathcal{N}_{l_t}\}}\sum_{r=1,k\in\mathcal{K}'}^{n_{bt}}\int_{\partial V_{tr}^k}\|q-p_{tr}\|^2\varphi(q)\frac{\partial\partial V_{tr}^k}{\partial p_{sm}}\mathcal{N}_{tr}^k\mathrm{d}q, \tag{4.15}$$

where $\mathcal{K}'=\{s|l_s\in\mathcal{N}_{l_t}\}$. Next, to calculate the derivative terms apperaing because of the dependency of the exterior boundaries on the nucleus, the chain rule can be applied as follows

$$\frac{\partial\partial V_{tr}^k}{\partial p_{sm}}=\frac{\partial l_s}{\partial p_{sm}}\frac{\partial\partial V_{tr}^k}{\partial l_s}, \tag{4.16}$$

where using (4.6) and (4.16), we obtain

$$\frac{\partial}{\partial p_{sm}}\Big(l_s\int_{V_s}\varphi(q)\mathrm{d}q\Big)=\frac{\partial}{\partial p_{sm}}\Big(\sum_{r=1}^{n_s}p_{sr}\int_{V_{sr}}\varphi(q)\mathrm{d}q\Big), \tag{4.17}$$

$$\frac{\partial l_s}{\partial p_{sm}}\Big(\big(\int_{V_s}\varphi(q)\mathrm{d}q\big)I_2+\big(\sum_{k\in\mathcal{K}'}\int_{\partial V_s}\varphi(q)\frac{\partial\partial V_s^k}{\partial l_s}\mathcal{N}_s^k\mathrm{d}q\big)l_s^\top\Big)= \\ \big(\int_{V_{sm}}\varphi(q)\mathrm{d}q\big)I_2+\sum_{r=1}^{n_s}\big(\int_{\partial V_{sr}}\varphi(q)\frac{\partial\partial V_{sr}}{\partial p_{sm}}\mathcal{N}_{sr}\mathrm{d}q\big)p_{sr}^\top. \tag{4.18}$$

The last term is again divided into two groups of the interior and boundary agents. Among the interior agents, only the Voronoi cells associated with agent located at p_{sm} and its neighbors are dependent on p_{sm}. From (4.18) and using the chain rule, we obtain

$$\frac{\partial l_s}{\partial p_{sm}}\Big(\big(\int_{V_s}\varphi(q)\mathrm{d}q\big)I_2+\big(\sum_{k\in\mathcal{K}'}\int_{\partial V_s}\varphi(q)\frac{\partial\partial V_s^k}{\partial l_s}\mathcal{N}_s^k\mathrm{d}q\big)l_s^\top- \\ \sum_{r=1,k\in\mathcal{K}'}^{n_{bs}}\big(\int_{\partial V_{sr}^k}\varphi(q)\frac{\partial\partial V_{sr}^k}{\partial l_s}\mathcal{N}_{sr}^k\mathrm{d}q\big)p_{sr}^\top\Big)=\big(\int_{V_{sm}}\varphi(q)\mathrm{d}q\big)I_2+ \\ \sum_{p_{sr}\in\{p_{sm},\mathcal{N}_{p_{sm}}\}}\big(\int_{\partial V_{sr}}\varphi(q)\frac{\partial\partial V_{sr}}{\partial p_{sm}}\mathcal{N}_{sr}\mathrm{d}q\big)p_{sr}^\top, \tag{4.19}$$

where n_{bs} is the number of the boundary agents in the s^{th} team. The effect of the moving boundaries with respect to the variation of the team nucleus is shown by the following notations

$$M_{\partial V_{sr}}^k=\int_{\partial V_{sr}^k}\varphi(q)\frac{\partial\partial V_{sr}^k}{\partial l_s}\mathcal{N}_{sr}^k\mathrm{d}q, \tag{4.20}$$

$$M_{\partial V_s}=\sum_{r=1,k\in\mathcal{K}'}^{n'_{bs}}M_{\partial V_{sr}}^k, \tag{4.21}$$

and the changing boundary of the interior agents due to the agents dynamics is represented by

$$M_{\partial V_{sr}}=\int_{\partial V_{sr}}\varphi(q)\frac{\partial\partial V_{sr}}{\partial p_{sm}}\mathcal{N}_{sr}\mathrm{d}q. \tag{4.22}$$

Using these notations results in the following

$$\frac{\partial l_s}{\partial p_{sm}}=\Big(M_{V_{sm}}I_2+\sum_{p_{sr}\in\{p_{sm},\mathcal{N}_{p_{sm}}\}}M_{\partial V_{sr}}p_{sr}^\top\Big)\Big(M_{V_s}I_2+M_{\partial V_s}l_s^\top-\sum_{r=1,k\in\mathcal{K}'}^{n'_{bs}}M_{\partial V_{sr}}^kp_{sr}^\top\Big)^{-1}. \tag{4.23}$$

The line that the points on the team boundaries belong to can be represented by

$$(\mathcal{N}_s^k)^\top (q - \frac{l_k + l_s}{2}) = 0, \quad q \in \partial V_s^k, \tag{4.24}$$

where ∂V_s^k is the boundary shared between teams s and k. The normal vector $\mathcal{N}_s^k$ associated with ∂V_s^k is obtained by

$$\mathcal{N}_s^k = \frac{l_k - l_s}{\|l_k - l_s\|}. \tag{4.25}$$

The partial derivative of (4.24) with respect to l_s is obtained as

$$\frac{\partial \mathcal{N}_s^k}{\partial l_s}(q - \frac{l_k + l_s}{2}) + (\frac{\partial \partial V_s^k}{\partial l_s} - \frac{1}{2})\mathcal{N}_s^k = 0, \tag{4.26}$$

where

$$\frac{\partial \mathcal{N}_s^k}{\partial l_s} = \frac{\mathcal{N}_s^k (\mathcal{N}_s^k)^\top - I_2}{\|l_k - l_s\|}. \tag{4.27}$$

Substituting (4.27) into (4.26), we have

$$\frac{\partial \partial V_s^k}{\partial l_s}\mathcal{N}_s^k = \frac{\mathcal{N}_s^k (\mathcal{N}_s^k)^\top - I_2}{\|l_k - l_s\|}(\frac{l_k + l_s}{2} - q) + \frac{1}{2}\mathcal{N}_s^k, \; q \in \partial V_s^k. \tag{4.28}$$

Because of the following equality that holds for the variation of the boundary edge ∂V_{tr}^k with respect to the variation of the nucleus l_s

$$\frac{\partial \partial V_{tr}^k}{\partial l_s}\mathcal{N}_{tr}^k = \frac{\partial \partial V_t^k}{\partial l_s}\mathcal{N}_t^k, \tag{4.29}$$

the derivative (4.15) can be rewritten as

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = \int_{V_{sm}} \frac{\partial}{\partial p_{sm}} \|q - p_{sm}\|^2 \varphi(q) \mathrm{d}q + \sum_{t \in \{s, g | l_g \in \mathcal{N}_{l_s}\}} \sum_{r=1, k \in \mathcal{K}'}^{n_{bt}} \int_{\partial V_{tr}^k} \|q - p_{tr}\|^2 \varphi(q) \frac{\partial l_s}{\partial p_{sm}} \frac{\partial \partial V_t^k}{\partial l_s} \mathcal{N}_t^k \mathrm{d}q. \tag{4.30}$$

Substituting (4.28) into (4.30) and the derivative of the first term lead to

$$\begin{aligned}
\frac{\partial \mathcal{G}}{\partial p_{sm}} = -2 \int_{V_{sm}} (q - p_{sm}) \varphi(q) \mathrm{d}q + \\
\frac{\partial l_s}{\partial p_{sm}} \Bigg(\sum_{r=1, k \in \mathcal{K}'}^{n_{bs}} \Bigg(\frac{\mathcal{N}_s^k (\mathcal{N}_s^k)^\top - I_2}{\|l_s - l_k\|} \int_{\partial V_{sr}^k} \|q - p_{sr}\|^2 \varphi(q) (\frac{l_k + l_s}{2} - q) \mathrm{d}q + \\
\frac{1}{2}\mathcal{N}_s^k \int_{\partial V_{tr}^k} \|q - p_{sr}\|^2 \varphi(q) \mathrm{d}q \Bigg) + \sum_{t \in \{g | l_g \in \mathcal{N}_{l_s}\}} \Bigg(\frac{I_2 - \mathcal{N}_t^s (\mathcal{N}_t^s)^\top}{\|l_t - l_s\|} \int_{\partial V_{tr}^s} \|q - p_{tr}\|^2 \varphi(q) (\frac{l_s + l_t}{2} - q) \mathrm{d}q + \\
\frac{1}{2}\mathcal{N}_t^s \int_{\partial V_{tr}^s} \|q - p_{tr}\|^2 \varphi(q) \mathrm{d}q \Bigg) \Bigg).
\end{aligned} \tag{4.31}$$

Further simplification of (4.31) using equations of mass and centroid (4.4) results in

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = -2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) + \left(M_{V_{sm}} I_2 + \sum_{p_{tr}\in\{p_{sm},\mathcal{N}_{p_{sm}}\}} M_{\partial V_{sr}} p_{sr}^{\top}\right)\left(M_{V_s} I_2 + M_{\partial V_s} l_s^{\top} - \sum_{r=1,k\in\mathcal{K}'}^{n_{bs}} M^k_{\partial V_{sr}} p_{sr}^{\top}\right)^{-1}$$
$$\left(\sum_{r=1,k\in\mathcal{K}'}^{n_{bs}} \left(\frac{\mathcal{N}_s^k(\mathcal{N}_s^k)^{\top} - I_2}{\|l_s - l_k\|}\int_{\partial V^k_{sr}} \|q - p_{sr}\|^2\varphi(q)(\frac{l_k + l_s}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_s^k \int_{\partial V^k_{tr}} \|q - p_{sr}\|^2\varphi(q)\mathrm{d}q\right) + \right.$$
$$\left.\sum_{t\in\{g|l_g\in\mathcal{N}_{l_s}\}} \left(\frac{I_2 - \mathcal{N}_t^s(\mathcal{N}_t^s)^{\top}}{\|l_t - l_s\|}\int_{\partial V^s_{tr}} \|q - p_{tr}\|^2\varphi(q)(\frac{l_s + l_t}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_t^s \int_{\partial V^s_{tr}} \|q - p_{tr}\|^2\varphi(q)\mathrm{d}q\right)\right). \tag{4.32}$$

As it can be seen, the boundaries of each team and its neighboring teams may vary with respect to the variation of the agent's location p_{sm}. This accounts for the sensitivity of the team boundaries with respect to any change in the nucleus of its associated team or the neighboring teams resulted from a change in agents' position. The term associated with the integration on the team boundaries in (4.32) is the same for all the agents in the t^{th} team. In fact, at every time step, it will be calculated just once for all the agents $r \in \{1, \ldots, n_{bt}\}$ with exterior boundaries shared with the neighboring teams.

4.2.3 Computation of the Voronoi cells

The Voronoi cells associated with each agent and team require a set of information to be computed online. As described before, the agents in the interior group are able to compute the Voronoi cell by communicating with the neighboring agents within the same team. However, the agents on the boundary group need the position of the nucleus of their team, as well as the nuclei of their neighboring teams. A greedy algorithm that provides the required data flow within each team and between the so-called leaders of different teams has been proposed in our recent work [3]. This is achieved at the cost of communicating with agents that are further away from each other; however, it allows the deployment of the agents without requiring the continuous communication among all the agents, which translates to less data exchange among the agents.

4.2.4 CONTROLLER DESIGN

A gradient decent-based control law is proposed here to guarantee the convergence of the teams of agents to their equilibrium point. The following dynamics is imposed on each agent

$$\dot{p}_{sm} = u_{sm} = \frac{K_{sm}}{2M_{V_{sm}}}\bigg(- \frac{\partial \mathcal{G}}{\partial p_{sm}}\bigg) = \frac{K_{sm}}{2M_{V_{sm}}}\bigg(2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) - \gamma_{sm}\bigg), \tag{4.33}$$

where $s = 1, \ldots, n,\ m = 1, \ldots, n_s$, K_{sm} is a positive scalar and

$$\begin{aligned}
\gamma_{sm} = &\bigg(M_{V_{sm}} I_2 + \sum_{p_{sr} \in \{p_{sm}, \mathcal{N}_{p_{sm}}\}} M_{\partial V_{sr}} p_{sr}^{\top}\bigg)\bigg(M_{V_s} I_2 + M_{\partial V_s} l_s^{\top} - \sum_{r=1, k \in \mathcal{K}'}^{n_{bs}} M_{\partial V_{sr}}^{k} p_{sr}^{\top}\bigg)^{-1} \\
&\bigg(\sum_{r=1, k \in \mathcal{K}'}^{n_{bs}} \bigg(\frac{\mathcal{N}_s^k (\mathcal{N}_s^k)^{\top} - I_2}{\|l_s - l_k\|} \int_{\partial V_{sr}^k} \|q - p_{sr}\|^2 \varphi(q)(\frac{l_k + l_s}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_s^k \int_{\partial V_{tr}^k} \|q - p_{sr}\|^2 \varphi(q)\mathrm{d}q\bigg) \\
&+ \sum_{t \in \{g | l_g \in \mathcal{N}_{l_s}\}} \bigg(\frac{I_2 - \mathcal{N}_t^s (\mathcal{N}_t^s)^{\top}}{\|l_t - l_s\|} \int_{\partial V_{tr}^s} \|q - p_{tr}\|^2 \varphi(q)(\frac{l_s + l_t}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_t^s \int_{\partial V_{tr}^s} \|q - p_{tr}\|^2 \varphi(q)\mathrm{d}q\bigg)\bigg).
\end{aligned} \tag{4.34}$$

By dividing K_{sm} by the varying term $2M_{V_{sm}}$, the control input is normalized to distribute the effect of both $2M_{V_{sm}}(C_{V_{sm}} - p_{sm})$ and γ_{sm} in the controller design. While the first term drives the agent towards its centroid, the second term is associated with the changing boundaries of the teams of agents. In other words, the above control law ensures that the agents are confined within the given dynamic boundaries of their team.

4.2.5 CONVERGENCE OF THE PROPOSED CONTROLLER

The proposed controller drives the agents to their optimum location while taking into account the moving boundaries. To ensure the convergence of the agents to their collective local optimum, the following lemma is proposed.

Lemma 4.2.1 *The agents with the assigned dynamics (4.33) converge to a local minimum by imposing the control law proposed in (4.33). That is,*

$$\lim_{\tau \to \infty} \big\| - 2M_{V_{sm}}(\tau)(C_{V_{sm}}(\tau) - p_{sm}(\tau)) + \gamma_{sm}(\tau)\big\| = 0,\ \forall s \in \{1, \ldots, n\}, \forall m \in \{1, \ldots, n_t\}. \tag{4.35}$$

Proof 4 *The asymptotic behavior cannot be proved through invoking standard invariant set theorems for time-varying systems. We hence use the Barbalat's lemma to show the asymptotic convergence of the system when each agent follows an optimal configuration. To this aim, the Lyapunov-like function associated with each team is defined as*

$$V = \mathcal{G}. \tag{4.36}$$

The derivative of this function is obtained as

$$\dot{V} = \sum_{s=1}^{n} \sum_{m=1}^{n_s} (\frac{\partial p_{sm}}{\partial \tau})^\top \frac{\partial \mathcal{G}}{\partial p_{sm}}. \tag{4.37}$$

Substituting (4.32) and (4.33) into (4.37), we obtain

$$\dot{V} = -\sum_{s=1}^{n} \sum_{m=1}^{n_t} \frac{K_{sm}}{2M_{V_{sm}}} \Big(-2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) + \gamma_{sm} \Big)^\top \Big(-2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) + \gamma_{sm} \Big). \tag{4.38}$$

Since $M_{V_{sm}}$ and K_{sm} are positive scalars, it can be concluded that the derivative (4.37) is non-positive, $\dot{V} \leq 0$. Due to the positivity of the cost function $\mathcal{G}$, it is also concluded that the Lyapunov-like function (4.36) is non increasing and hence lower bounded. As shown in [125], $\ddot{V}(\tau)$ is uniformly bounded that results in the uniform continuity of $\dot{V}(\tau)$. Next, since $V(\tau)$ is bounded and due to the continuity of $\dot{V}(\tau)$, it is proven by Barbalat's lemma that

$$\lim_{\tau \to \infty} \dot{V} = 0 \quad \Longrightarrow \quad \lim_{\tau \to \infty} \| -2M_{V_{sm}}(\tau)(C_{V_{sm}}(\tau) - p_{sm}(\tau)) + \gamma_{sm}(\tau) \| = 0, \tag{4.39}$$

for $\forall s \in \{1, \ldots, n\}, \forall m \in \{1, \ldots, n_t\}$.

Due to the dependency of the nucleus of the t^{th} team on the position of its agents, $l_t(p_{t1}, \ldots, p_{tm})$, Theorem 4.2.2 below guarantees the convergence of the nucleus of teams.

Theorem 4.2.2 *According to (4.6) and the dynamics of the nuclei l_t, it can be inferred from Lemma 4.2.1 that the nuclei of the agents will also converge to the local minimum of the team level optimization problem with the cost function defined as in (4.7).*

Proof 5 *The dynamics imposed on the nuclei of the teams are indirectly resulted from the dynamics obtained from (4.33). This can be seen by taking the time derivative of (4.6) as*

$$\frac{\partial l_s}{\partial \tau} = \sum_{m=1}^{n_s} \frac{\partial l_s}{\partial p_{sm}} \dot{p}_{sm}. \tag{4.40}$$

Substituting (4.23) into (4.40), we obtain

$$\frac{\partial l_s}{\partial \tau} = \sum_{m=1}^{n_s} \Bigg(M_{V_{sm}} I_2 + \sum_{p_{sr} \in \{p_{sm}, \mathcal{N}_{p_{sm}}\}} M_{\partial V_{sr}} p_{sr}^{\top} \Bigg) \Bigg(M_{V_s} I_2 + M_{\partial V_s} l_s^{\top} - \sum_{r=1, k \in \mathcal{K}'}^{n_{bs}} M_{\partial V_{sr}}^{k} p_{sr}^{\top} \Bigg)^{-1} \dot{p}_{sm}. \tag{4.41}$$

It can be seen that when the agents inside a team converge to their associated local minima according to Lemma 4.2.1, the nuclei of the teams will also asymptotically converge to their local minima as

$$\lim_{\tau \to 0} \Bigg\| \sum_{m=1}^{n_s} \Bigg(M_{V_{sm}}(\tau) I_2 + \sum_{p_{sr} \in \{p_{sm}, \mathcal{N}_{p_{sm}}\}} M_{\partial V_{sr}}(\tau) p_{sr}^{\top}(\tau) \Bigg) \Bigg(M_{V_s}(\tau) I_2 + M_{\partial V_s}(\tau) l_s^{\top}(\tau) - \sum_{r=1, k \in \mathcal{K}'}^{n_{bs}} M_{\partial V_{sr}}^{k}(\tau) p_{sr}^{\top}(\tau) \Bigg)^{-1} \dot{p}_{sm}(\tau) \Bigg\| = 0. \tag{4.42}$$

4.3 Formation Control of the Teams of Agents

The idea of team-based partitioning relies on forming teams of agents with respect to their capabilities and dynamics. It is highly likely that agents require to take different formations within their confined region while performing the assigned coverage task(s). A major factor in changing the formation of the agents is to change their relative distances. We introduce a formation term within each team while aiming at achieving the main partitioning task. The following cost function is defined to ensure that the agents can maintain a certain formation throughout their mission

$$\mathcal{G}_t(\mathcal{P}_t, \mathcal{Q}_t) = \sum_{m=1}^{n_t} \big(\int_{V_{tm}} \|q - p_{tm}\|^2 \varphi(q) \mathrm{d}q + \alpha_{tm} \|p_{tm} - l_t\|^2 \big), \tag{4.43}$$

in which the formation factor α_{tm} is a positive scalar. To solve the associated optimization problem, the derivative of (4.43) is obtained as

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = \int_{V_{sm}} \frac{\partial}{\partial p_{sm}} \|q - p_{sm}\|^2 \varphi(q) \mathrm{d}q + \sum_{t \in \{s, g | l_g \in \mathcal{N}_{l_s}\}} \sum_{r=1, k \in \mathcal{K}'}^{n_{bt}} \int_{\partial V_{tr}^k} \|q - p_{tr}\|^2 \varphi(q) \frac{\partial l_s}{\partial p_{sm}} \frac{\partial \partial V_t^k}{\partial l_s} \mathcal{N}_t^k \mathrm{d}q + 2\alpha_{sm} (I_2 - \frac{\partial l_s}{\partial p_{sm}})(p_{sm} - l_s). \tag{4.44}$$

Solving this optimization problem follows the same lines as those in the previous section except that there is now an extra term representing the formation of the agents. Employing (4.32) and

(4.34), we have

$$\frac{\partial \mathcal{G}}{\partial p_{sm}} = -2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) + \gamma_{sm} +$$
$$2\alpha_{sm}\Bigg(I_2 - \big(M_{V_{sm}} I_2 + \sum_{p_{tr}\in\{p_{sm},\mathcal{N}_{p_{sm}}\}} M_{\partial V_{tr}} p_{tr}^{\top}\big)\big(M_{V_s} I_2 + M_{\partial V_s} l_s^{\top} - \sum_{r=1,k\in\mathcal{K}'}^{n_{bs}} M_{\partial V_{sr}}^{k} p_{sr}^{\top}\big)^{-1}\Bigg)(p_{sm} - l_s). \tag{4.45}$$

The gradient decent method results in the following expression for the agents dynamics assigned to maintain a certain formation through the given formation factor

$$\dot{p}_{sm} = u_{sm} = K_{sm}\Bigg(-\frac{\partial \mathcal{G}}{\partial p_{sm}}\Bigg) = K_{sm}\Bigg(2M_{V_{sm}}(C_{V_{sm}} - p_{sm}) - \gamma_{sm} - 2\alpha_{sm}\Lambda_{sm}(p_{sm} - l_s)\Bigg), \tag{4.46}$$

where

$$\Lambda_{sm} = I_2 - \big(M_{V_{sm}} I_2 + \sum_{p_{tr}\in\{p_{sm},\mathcal{N}_{p_{sm}}\}} M_{\partial V_{tr}} p_{tr}^{\top}\big)\big(M_{V_s} I_2 + M_{\partial V_s} l_s^{\top} - \sum_{r=1,k\in\mathcal{K}'}^{n_{bs}} M_{\partial V_{sr}}^{k} p_{sr}^{\top}\big)^{-1}. \tag{4.47}$$

Remark 4.4 *The convergence of the agents position to their local optima by using the proposed formation control law in (4.46) and (4.47) is ensured by considering the Lyapunov function as $V = \mathcal{G}$. The proof follows the same lines as those in Theorem 4.2.2.*

Remark 4.4 implies that the agents converge to their associated local minima while maintaining a desired formation. In fact, changing the formation factor can change the relative distance of the agents from their respective nuclei resulting in a different formation of the agents within each team. As an extension of the present work and in a more generic scenario, the time-varying formation factor might be considered to represent any change in the formation due to, e.g., the changing environment or the assigned tasks.

4.4 Simulation Results and discussion

To examine the capabilities of the proposed approaches, the team-based partitioning is validated using numerical examples. First, the proposed team-based partitioning is used to deploy four teams each consisting of five agents. The designed algorithm deploys the teams of agents in the main

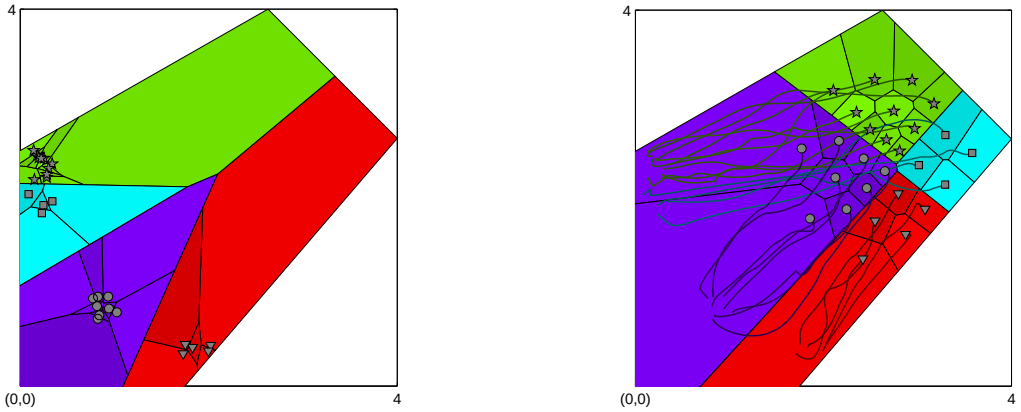


Figure 4.2: Initial (left) and final configurations (right) for 4 teams of five agents; each marker represents the agents belonging to different teams.

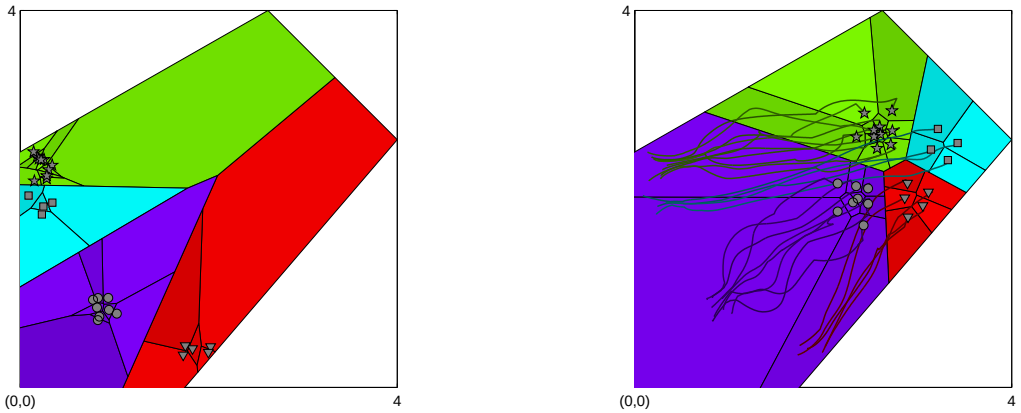


Figure 4.3: Initial (left) and final configurations (right) for 4 teams of five agents maintaining the desired flocking formation for agents belonging to different teams.

region while optimizing the underlying deployment problem within each team. As seen from Figure 4.2, the proposed approach can successfully deploy the teams of agents in order to obtain the optimal coverage on the given area. The importance function is considered to be

$$\varphi(x, y) = \exp(-\frac{(x-3)^2 + (y-2.5)^2}{2\sigma^2}), \tag{4.48}$$

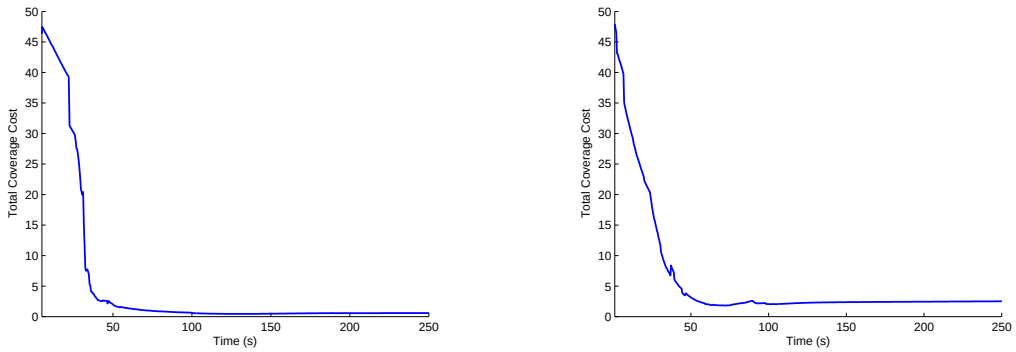


Figure 4.4: The convergence of the total coverage cost function illustrating the local optimum solution to the team based deployment (left) and the team based formation control (right).

where the variance is $\sigma^2 = 0.3$, and the controller parameter chosen as $K_{sm} = 0.012$. As expected, the coverage cost function decreases through the deployment process. Figure 4.4 (left) demonstrates the convergence of the cost function over time.

Next, to examine the effectiveness of the proposed formation control, a constant formation coefficient is chosen for all the teams. The teams of agents and their final formation is shown in Figure 4.3, where agents take a specific formation within the Voronoi of their associated team to fulfill the desired objective such as a required communication radius to ensure a consistent communication for agents inside each team or any other formation-related task. The controller parameter and the formation control coefficient are chosen as $K_{sm} = 0.015$ and $\alpha_{sm} = 2.5$, respectively. The convergence of the total cost function is shown in Figure 4.4 (right). As expected, the desired formation comes with a higher coverage cost. This can be seen in the final values of the two convergence plots in Figure 4.4, where the cost value in the formation control is higher.

Another numerical example is illustrated here to study a more generic case where each team takes a different formation due to its assigned task. This can be seen as a potential representation of heterogeneity when deploying a number of agents that pursue different objectives because of their coverage capabilities. The configuration shown in Figure 4.5 is achieved by choosing different formation coefficients α_{tm} for each agent within each team. The gradient-decent based control

Table 4.1: Formation coefficients for the four teams of agents shown in Figure 4.5.

α_{11}	α_{12}	α_{13}	α_{14}	α_{15}				
0.6	0.6	0.9	4	0.6				
α_{31}	α_{32}	α_{33}	α_{34}	α_{35}	α_{36}			
0.8	3	3	0.9	5	0.8			
α_{41}	α_{42}	α_{43}	α_{44}	α_{45}	α_{46}	α_{47}	α_{48}	
0.3	0.3	1	0.9	1	0.3	0.5	0.9	
α_{21}	α_{22}	α_{23}	α_{24}	α_{25}	α_{26}	α_{27}	α_{28}	α_{29}
0.7	0.7	0.2	0.7	0.2	0.2	0.2	0.2	0.2

gain is chosen as $K_{sm} = 0.01$. Table 4.1 shows the formation coefficients of each agent belonging to different teams. The agents tend to converge to a certain formation due to the relative impact of the associated coverage and formation terms. This relative impact is controlled by choosing a relatively lower or higher formation coefficient to enforce the agents to stay closer to or further away from the nucleus of their team. The higher formation term implies a stronger tendency to bring the agents towards the nucleus of the team by increasing the formation coefficient while the coverage term attempts to scatter them over the teams' associated Voronoi.

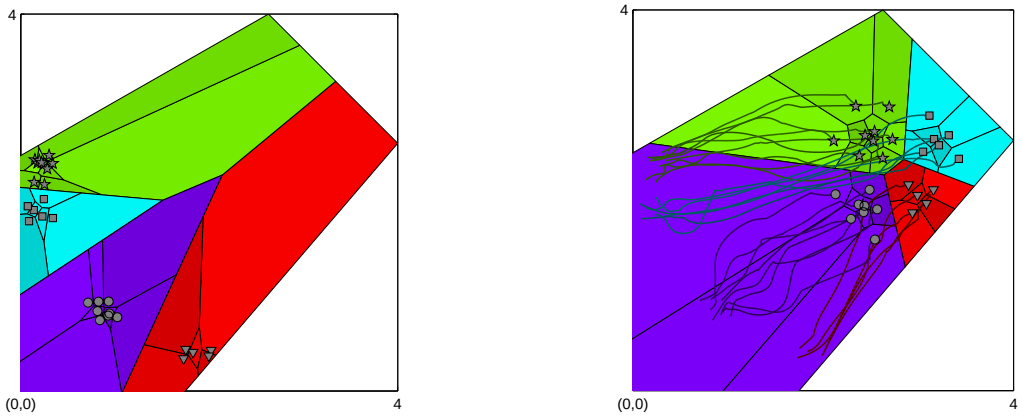


Figure 4.5: Initial (left) and final configurations (right) for four teams of 5 (asterisk), 9 (star), 6 (square) and 8 (circle) agents, where each marker represents the agents belonging to different teams.

4.5 Concluding Remarks

To adapt to the complexity of the coverage problems in real world applications, a team-based coverage approach is presented in this paper. The proposed approach solves the problem in a more local way leading to the partitioning of the main region into multiple subregions associated with multiple deployed teams. This is beneficial to handle tasks that require a more diverse group of robots due to their complexity. Due to the diversity of the coverage problems, the need for deploying robots with different dynamics or communication characteristics might be inevitable, and hence, the proposed team-based approach is an attempt to modify and improve existing methods in a way that the teams of robots can divide the region among themselves based on their capabilities. This provides the means for autonomous deployment of multiple teams of robots when there is a need for different types of robots form both communication and dynamics perspectives.

CHAPTER 5

COVERAGE CONTROL OF MOVING SENSOR NETWORKS IN SURFACE FLOW FIELDS [1]

[1] F. Abbasi, A. Mesbahi and J. Mohammadpour: Coverage Control of Moving Sensor Networks in Surface Flow Fields. 2016. *Submitted to IEEE Transactions on Control Systems Technology.*

ABSTRACT

This paper addresses the coverage problem in surface flow fields, where it is desired to cover a long region by moving within the boundaries of the flow stream. To this purpose, a group of autonomous mobile sensors are deployed aiming to minimize a sensing cost function. The coverage area, considered to be a region with changing boundaries, is directed to move along the boundaries of the flow until it reaches to the final destination. Throughout this process, the agents adapt to the varying coverage area by imposing the dynamics of the boundaries on their respective control law. The presented control law ensures that the agents move toward the centroid of their respective Voronoi cell while taking into account the effect of the moving boundaries. The proposed coverage scheme is examined via two numerical examples that use sections of Colorado and Ohio rivers. The proposed algorithm deploys the agents within the boundaries of the river and ensures the optimum partitioning for the moving coverage area.

5.1 INTRODUCTION

There have been advancements on developing techniques for deployment of a group of agents in a given environment to perform assigned distributed tasks [54, 55, 110]. The core problem can be seen as a workload sharing task to assign the share of each agent from the total workload. The underlying optimization problem is NP-hard (Non-deterministic Polynomial-time hard) [46], and hence, finding a local minimum of the cost function is desired. The Voronoi partitions and the Lloyd's algorithm are proposed as solutions to the coverage problem [83]. The centroid of agents' Voronoi cells is considered as the optimal sensor location at each step to design decentralized control laws [32]. In this strategy, the given environment is partitioned into Voronoi subregions and each agent drives toward the centroid of its Voronoi cell. The existing methods are developed to deploy agents in a given and invariant environment and may not address the case where for various reasons the coverage area has dynamics.

The fresh water resources such as rivers are the main sources for supplying water for daily usage and irrigation that are also used for recreational activities. The rivers can be negatively impacted by various factors, including human activities, heavy metals, nitrogen, phosphorus and acid mine [107]. Monitoring of the quality of the surface water in rivers plays a critical rule in conserving water resources [107, 128]. Several monitoring methods have been introduced, e.g., in [107, 128] based on human activities or monitoring stations. However, the existing methods are not suitable for real-time monitoring of a long section of large rivers [40]. Recently, autonomous mobile sensors have been employed for real-time monitoring and data collection in rivers [7, 26, 63, 98]. In [39, 41], an autonomous surface vehicle (ASV) is used for measuring a range of water quality properties and greenhouse gas emissions while avoiding obstacles.

In several real world applications, a non-autonomous network of mobile sensors are employed to monitor flow fields while moving along the streamline [34, 77]. A proximity metric of generalized Voronoi diagram in a flow field has been used in [137] to compute the shortest time to move from one point to another in the presence of constant drift. However, the computation of the Voronoi diagram is difficult due to the existence of singularities in the equi-distant curves [92, 93]. As it is expected from the flow dynamics, it is more efficient due to the preserved energy consumption when the mobile sensors do not move in the opposite direction of the flow in the coverage problem [69]. Additionally, the available actuation of mobile sensors is limited and might not compensate for the opposite flow velocity to generate the desired speed in fast flow fields [71]. To avoid this, a group of mobile sensors are deployed to cover the maximum area in a fixed time using Snell's law of refraction while they cannot move in the opposite of the flow direction [69, 70].

To solve the coverage problem in constant flow fields by taking into account both energy consumption and traveling time, a refined approximated Voronoi diagram is used to deploy a group of agents in [123]. Even though the presented approach is a detailed study that considers the energy factor, it does not account for the changing flow fields where it is needed to move along the flow while maintaining optimal coverage. Although it is assumed that the flow is constant along the

streamline in [69, 70, 123], it may change along the environment in many real world applications [142, 144, 148].

In this paper, we study the coverage control problem for continuous and potentially long flow fields. In this work, it is desired to move along the flow direction to preserve energy consumption; however, by providing enough power the presented method can also be implemented to move upstream. This is motivated by the fact that an efficient coverage strategy should use properties of the flow fields, such as the velocity and the direction of the flow. Therefore, it would be possible to improve reliability and energy consumption of the deployment algorithm.

Throughout this paper, the flow field is modeled by two curved lines indicating the boundaries of any flow stream as river borders. Our objective is to cover whole environment by deploying agents into a moving region in the flow field. The agents move within the boundaries of the river and converge to an optimum configuration within their respective coverage area and are able to change their area by expanding or compressing due to changing flow conditions. The speed of the coverage region along the flow determines the collective speed of the agents within their region. Two virtual guidance points are introduced to control the dynamics of the moving covered region. The imposed dynamics on the guidance points affect the boundaries of the associated moving coverage region that in turn changes the dynamics of the agents. These guidance points are located in the back and front areas of the moving region which respectively determine the front and the rear boundaries of the group of agents while moving along the flow field. As an immediate use of these points, one can assign different dynamics to these points to achieve a compressed or expanded coverage area.

The remainder of this paper is structured as follows. Definitions and the problem statement are provided for the coverage control for a flow field in Section II. Section III introduces an approach for optimal voronoi partitioning and controller Design. Section IV presents numerical simulation results to illustrate the optimal coverage for flow type problems.

5.1.1 NOTATIONS

We use $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_+$ to respectively denote the sets of natural, real, and nonnegative real numbers. Throughout the paper, I_r denotes $r \times r$ identity matrix. We define $\mathcal{Q}$ as a polytope in $\mathbb{R}^2$ and let $\mathscr{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_t\}$ be a *partition* of $\mathcal{Q}$ as a collection of t closed subsets with disjoint interiors. The boundary of $\mathcal{Q}$ is denoted by $\partial\mathcal{Q}$. Moreover, the so-called *distribution density function* is denoted by φ where $\varphi : \mathcal{Q} \to \mathbb{R}_+$ represents the probability of some phenomenon occurring over space $\mathcal{Q}$. The function φ is assumed to be measurable and absolutely continuous. The Euclidean distance function is denoted by $\|\cdot\|$, and $|\mathcal{Q}|$ represents the Lebesgue measure of the convex subset $\mathcal{Q}$. The vector set $\mathcal{P}_t = (p_{t1}, p_{t2}, \ldots, p_{tn_t})$ is the location of n_t agents belonging to t^{th} team.

5.2 UNDERLYING OPTIMIZATION PROBLEM

In the coverage control literature, the *locational cost function* is defined as a measure of the sensing performance. In this work, we present a modified version of the locational function that is suitable for the proposed coverage method for n number of agents belonging to P, which is the set of all agents. Assuming that i^{th} agent is assigned to the region $\mathcal{W}_i$, the objective is to minimize the sensing cost function by finding the optimal locations of the agents and their assigned regions $\mathcal{W}_i$ whose union is $\mathcal{Q}$. In the flow framework, it is desired to keep the sensing performance as high as possible while the region associated with agent is changing due to the changing boundaries of the main region. The sensing performance is evaluated as a function of the distance from the agent, i.e., $f(\|q - p_i\|)$, where $q \in \mathcal{Q}$. In this context, the density function is set to be constant to represent the uniform coverage and all the agents are assigned over the moving space along the given area. The dynamic partitioning approach is introduced in this paper to address the problem of the coverage for flow related problems, where the agents need to move along the boundaries of the region to perform the desired coverage task.

5.2.1 VORONOI PARTITIONS

The main objective of this work is to adopt a dynamic coverage area within the agents deployment and partitioning framework. To achieve this, we need to define an optimization problem that accounts for the changing boundaries of the associated region in a way that it ensures the optimal coverage for a number of moving agents. This optimization problem should consider the agents, individual cost function defined within the dynamic boundaries of the region associated with all the agents. We first define the collective position of the team by the centroid l as a function of the agents position in the associated team, i.e., $l = g(p_1, p_2, \ldots, p_n)$. The dependency of l on the position of the agents is discussed later. Next, we partition the changing polytope $\mathcal{Q}$ into a set of Voronoi cells $\mathcal{V}_t(\mathcal{P}) = \{V_1, V_2, \ldots, V_n\}$ generated by the agents $(p_1, p_2, \ldots, p_n)$ as

$$V_m = \{q \in \mathcal{Q} | \ \|q - p_m\| \leq \|q - p_r\|, r = 1, \ldots, n, r \neq m\}, \tag{5.1}$$

where p_m denotes the location of m^{th} agent for $m \in \{1, \ldots, n\}$.

Depending on the position of the agents, they might require a different set of data, i.e., their neighbors' position, to be maintained. Some agents share boundaries only with other agents while the agents on the boundaries of the region are neighbors not only with other agents but they may also share edges with the polytope $\mathcal{Q}$. In general, the edges associated with each agent ∂V_m are either an edge shared with other agents or edges shared with the dynamic region of the team of agents depending on the position of the agent within the team. An edge that is shared with the neighboring agent f in the same team is shown by $\partial V_{m,f}$. The edges associated with the agents on the boundaries that are shared with the main polytope $\mathcal{Q}$, the boundaries of the rear and front lines confining the region are represented by ∂V_m^B, ∂V_m^F and ∂V_m^R, respectively. Also, ∂V_1^B and ∂V_2^B indicate the fixed boundaries of the region forming the main stream of the flow that are shown by smooth splines. Figure 5.1 illustrates the boundaries and their normal vectors for Voronoi V_m. We recall the basic characteristics of the Voronoi partitions including their associated mass, centroid,

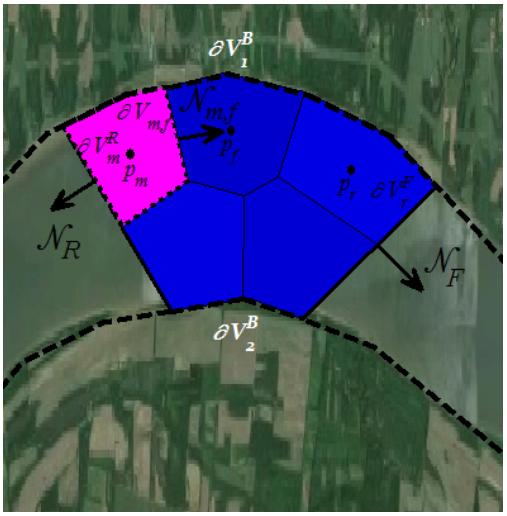


Figure 5.1: An illustrative example of the Voronoi partitioning in a flow field, where the boundaries and their associated normal vectors for the Voronoi cell V_m are shown for the edges shared with f^{th} agent, in addition to the edge ∂V_m^B shared with the main region $\mathcal{Q}$.

and polar moment of inertia defined as [32]

$$\begin{aligned} M_{V_m} &= \int_{V_m} \varphi(q)\mathrm{d}q, \quad C_{V_m} = \frac{1}{M_{V_m}} \int_{V_m} q\ \varphi(q)\mathrm{d}q, \\ J_{V_m,p_m} &= \int_{V_m} \|q - p_m\|^2 \varphi(q)\mathrm{d}q. \end{aligned} \tag{5.2}$$

The centroid of the team that is a function of the agents position is defined as

$$l = \frac{\sum_{m=1}^{n} M_{V_m} p_m}{\sum_{m=1}^{n} M_{V_m}}. \tag{5.3}$$

As described earlier, the centroid l is a representative of the agents position in the team and can be considered as the collective position of the agents that is needed for the computation of the dynamic boundaries of the region $\mathcal{Q}$.

5.2.2 Formulation of the local optimization problem

The deployment task in the proposed flow framework can be addressed by solving an optimization problem over the polytope $\mathcal{Q}$ where the team's region needs to be optimized in the sense of its

associated sensing function. Hence, the following cost function is considered

$$\mathcal{G}(\mathcal{P},\mathscr{Q}) = \int_{\mathcal{Q}_R} \|q-p_R\|^2\varphi(q)\mathrm{d}q + \sum_{m=1}^{n}\int_{\mathcal{Q}_m} \|q-p_m\|^2\varphi(q)\mathrm{d}q + \int_{\mathcal{Q}_F} \|q-p_F\|^2\varphi(q)\mathrm{d}q, \quad (5.4)$$

where $\mathcal{G}$ is the collective cost function associated with the sensing performance of the agents in the set $\mathcal{P}$. Also, p_F and p_R represent the front and rear guidance points, respectively. The sensing performance is considered as $f(\|q-p_m\|) = \|q-p_m\|^2$ for the m^{th} agent. The solution to (5.4) gives a local minimum to the deployment problem where agents are considered as the members of a collaborating team sweeping through the region. We define a set of polygons as $\mathscr{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_n\}$, with disjoint interiors, whose union is $\mathcal{Q}$.

Remark 5.1 *It is proven that among different partitioning schemes, the Voronoi partitions are optimum in the sense of minimizing the defined cost function (5.4) [3, 32]. Hence, for a given set of agents with position $\mathcal{P} \in \mathcal{Q}$ and a partition $\mathscr{Q}$ of $\mathcal{Q}$, we have*

$$\mathcal{G}(\mathcal{P},\mathcal{V}(\mathcal{P})) \leq \mathcal{G}(\mathcal{P},\mathscr{Q}), \quad (5.5)$$

which implies that the Voronoi cells represent the optimum partitioning of the area associated with dynamic region of the agents.

We first need to define the dynamic boundaries of the region $\mathcal{Q}$. For this purpose, two virtual points l_R and l_F called guidance points are assigned before and after the group of agents, respectively. The locations of these points are assumed to be on the trajectory representing the mean of the fixed side boundaries of the main region that is obtained by $\partial V^a = \frac{\partial V_1^B + \partial V_2^B}{2}$. In other words, these points always have the same distance from the fixed boundaries. The lines representing the dynamic boundaries of the region that confine the coverage area are described by

$$(\mathcal{N}_i)^\top (q - \frac{l+p_i}{2}) = 0, \quad q \in \partial V_r^i, \quad (5.6)$$

where $i \in \{F, R\}$ and the normal vector $\mathcal{N}_i$ associated with ∂V_r^i is obtained by

$$\mathcal{N}_i = \frac{p_i - l}{\|p_i - l\|}. \quad (5.7)$$

The dynamics on the guidance points dictate the speed and direction in which the sensing coverage is carried out on the given area. The next step is to obtain (locally optimum) location of the agents. The derivative of the cost function (5.4) associated with n agents is

$$\frac{\partial \mathcal{G}}{\partial p_m} = \frac{\partial}{\partial p_m}\Bigg(\int_{Q_R} \|q - p_R\|^2 \varphi(q) \mathrm{d}q + \sum_{r=1}^{n} \int_{Q_r} \|q - p_r\|^2 \varphi(q) \mathrm{d}q + \int_{Q_F} \|q - p_F\|^2 \varphi(q) \mathrm{d}q \Bigg), \quad m = 1, \ldots, n. \tag{5.8}$$

The solution to this optimization problem differs from that to the conventional sensing cost functions due to the previously defined dependency of the edges of the agents shared with the boundaries on imposed dynamics of the guidance points. The derivative with respect to the coordinate of agent p_m is obtained as

$$\frac{\partial \mathcal{G}}{\partial p_m} = \int_{V_m} \frac{\partial}{\partial p_m} \|q - p_m\|^2 \varphi(q) \mathrm{d}q + \Bigg(\sum_{r=1}^{n} \int_{\partial V_r} \|q - p_r\|^2 \varphi(q) \frac{\partial \partial V_r}{\partial p_m} \mathcal{N}_r \mathrm{d}q + \int_{\partial V^F} \|q - p_F\|^2 \varphi(q) \frac{\partial \partial V^F}{\partial p_m} \mathcal{N}_F \mathrm{d}q + \int_{\partial V^R} \|q - p_R\|^2 \varphi(q) \frac{\partial \partial V^R}{\partial p_m} \mathcal{N}_R \mathrm{d}q \Bigg), \tag{5.9}$$

where $\partial V^F = \partial \mathcal{Q} \cap \partial \mathcal{Q}_F$ and $\partial V^R = \partial \mathcal{Q} \cap \partial \mathcal{Q}_R$. It can be inferred from the proposed partitioning that the boundaries of the Voronoi cell V_r are either directly or indirectly dependent on p_m. The direct dependency is obviously resulted from the definition of the Voronoi cells (5.1). We note that V_m and the voronoi cells in its neighborhood are directly dependent on p_m. The indirect dependency can be seen in the shared boundaries with the guidance agents, where they share at least one edge with the guidance agents (or contribute at least one edge to the boundary of the group of agents).

Remark 5.2 *The integral on each boundary shared with the neighboring agents is identical for agents on both sides except that the normals will have opposite signs, i.e., $\mathcal{N}_{m,f} = -\mathcal{N}_{f,m}$. Hence, we have*

$$\sum_{f \in \mathcal{F}} \int_{\partial V_{m,f}} \|q - p_m\|^2 \varphi(q) \frac{\partial \partial V_{m,f}}{\partial p_m} \mathcal{N}_{m,f} dq = -\sum_{f \in \mathcal{F}} \int_{\partial V_{f,m}} \|q - p_r\|^2 \varphi(q) \frac{\partial \partial V_{f,m}}{\partial p_m} \mathcal{N}_{f,m} dq, \tag{5.10}$$

where $\mathcal{N}_{f,m}$ is the normal vector associated with the edge $\partial V_{f,m}$, and $\mathcal{F} = \{f | p_f \in \mathcal{N}_{p_m}\}$ with $\mathcal{N}_{p_m}$ representing the set of agents that share boundaries with agent p_m. According to Remark 5.2, the terms associated with shared boundaries with other neighbouring agents will cancel out and we have

$$\frac{\partial \mathcal{G}}{\partial p_m} = \int_{V_m} \frac{\partial}{\partial p_m} \|q - p_m\|^2 \varphi(q) \mathrm{d}q + \Big(\sum_{r=1, i \in \{F,R\}}^{n_b} \int_{\partial V_r^i} \|q - p_r\|^2 \varphi(q) \frac{\partial \partial V_r^i}{\partial p_m} \mathcal{N}_i \mathrm{d}q + \int_{\partial V^F} \|q - p_F\|^2 \varphi(q) \frac{\partial \partial V^F}{\partial p_m} \mathcal{N}_F \mathrm{d}q + \int_{\partial V^R} \|q - p_R\|^2 \varphi(q) \frac{\partial \partial V^R}{\partial p_m} \mathcal{N}_R \mathrm{d}q \Big), \quad (5.11)$$

where n_b is the number of agents sharing edges with $\partial \mathcal{Q}_F$ and $\partial \mathcal{Q}_R$. Next, to calculate the derivative terms appearing because of the dependency of the shared edges with dynamic boundaries on the nucleus, the chain rule can be applied as follows

$$\frac{\partial \partial V_r^i}{\partial p_m} = \frac{\partial l}{\partial p_m} \frac{\partial \partial V_r^i}{\partial l}, \quad (5.12)$$

where using (5.3) and (5.12), we obtain

$$\frac{\partial}{\partial p_m} \Big(l \int_{\mathcal{Q}} \varphi(q) \mathrm{d}q \Big) = \frac{\partial}{\partial p_m} \Big(\sum_{r=1}^{n} p_r \int_{V_r} \varphi(q) \mathrm{d}q \Big), \quad (5.13)$$

$$\frac{\partial l}{\partial p_m} \Big(\big(\int_{\mathcal{Q}} \varphi(q) \mathrm{d}q \big) I_2 + \big(\sum_{i \in \{F,R\}} \int_{\partial \mathcal{Q}} \varphi(q) \frac{\partial \partial V^i}{\partial l} \mathcal{N}_i \mathrm{d}q \big) l^\top \Big) = \big(\int_{V_m} \varphi(q) \mathrm{d}q \big) I_2 + \sum_{r=1}^{n} \big(\int_{\partial V_r} \varphi(q) \frac{\partial \partial V_r}{\partial p_m} \mathcal{N}_r \mathrm{d}q \big) p_r^\top. \quad (5.14)$$

The last term can be discussed in two groups of edges; the ones shared with neighboring agents and the shared edges with dynamic boundaries. It should be noted that only the Voronoi cells associated with the agent located at p_m and its neighbors are dependent on p_m. From (5.14) and using the chain rule, we obtain

$$\frac{\partial l}{\partial p_m} \Big(\big(\int_{\mathcal{Q}} \varphi(q) \mathrm{d}q \big) I_2 + \big(\sum_{i \in \{F,R\}} \int_{\partial \mathcal{Q}} \varphi(q) \frac{\partial \partial V^i}{\partial l} \mathcal{N}_i \mathrm{d}q \big) l^\top - \sum_{r=1, i \in \{F,R\}}^{n_b} \big(\int_{\partial V_r^i} \varphi(q) \frac{\partial \partial V_r^i}{\partial l} \mathcal{N}_i \mathrm{d}q \big) p_r^\top \Big) = \big(\int_{V_m} \varphi(q) \mathrm{d}q \big) I_2 + \sum_{p_r \in \{p_m, \mathcal{N}_{p_m}\}} \big(\int_{\partial V_r} \varphi(q) \frac{\partial \partial V_r}{\partial p_m} \mathcal{N}_r \mathrm{d}q \big) p_r^\top. \quad (5.15)$$

The effect of the moving boundaries with respect to the variation of the centroid of the group of agents is shown by the following notations

$$M^i_{\partial V_r} = \int_{\partial V^i_r} \varphi(q) \frac{\partial \partial V^i_r}{\partial l} \mathcal{N}_i \mathrm{d}q, \tag{5.16}$$

$$M_{\partial \mathcal{Q}} = \sum_{r=1, i\in\{F,R\}}^{n_b} M^i_{\partial V_r}, \tag{5.17}$$

and the changing boundary of the interior agents due to the agents dynamics is represented by

$$M_{\partial V_r} = \int_{\partial V_r} \varphi(q) \frac{\partial \partial V_r}{\partial p_m} \mathcal{N}_r \mathrm{d}q. \tag{5.18}$$

Using these notations results in the following

$$\frac{\partial l}{\partial p_m} = \left(M_{V_m} I_2 + \sum_{p_r \in \{p_m, \mathcal{N}_{p_m}\}} M_{\partial V_r} p_r^\top \right) \left(M_Q I_2 + M_{\partial Q} l^\top - \sum_{r=1, i\in\{F,R\}}^{n_b} M^i_{\partial V_r} p_r^\top \right)^{-1}. \tag{5.19}$$

The partial derivative of (5.6) with respect to l is obtained as

$$\frac{\partial \mathcal{N}_i}{\partial l}(q - \frac{p_i + l}{2}) + (\frac{\partial \partial V^i_s}{\partial l} - \frac{1}{2})\mathcal{N}^i = 0, \tag{5.20}$$

where

$$\frac{\partial \mathcal{N}_i}{\partial l} = \frac{\mathcal{N}_i (\mathcal{N}_i)^\top - I_2}{\|p_i - l\|}. \tag{5.21}$$

Substituting (5.21) into (5.20), we have

$$\frac{\partial \partial V^i_s}{\partial l} \mathcal{N}_i = \frac{\mathcal{N}_i (\mathcal{N}_i)^\top - I_2}{\|p_i - l\|} (\frac{p_i + l}{2} - q) + \frac{1}{2} \mathcal{N}_i, \ q \in \partial V^i_s. \tag{5.22}$$

Because of the following equality that holds for the variation of the boundary edge ∂V^i_r with respect to the variation of the nucleus l

$$\frac{\partial \partial V^i_r}{\partial l} \mathcal{N}_i = \frac{\partial \partial V^i}{\partial l} \mathcal{N}_i, \tag{5.23}$$

the derivative (5.9) can be rewritten as

$$\begin{aligned} \frac{\partial \mathcal{G}}{\partial p_m} = \int_{V_m} \frac{\partial}{\partial p_m} \|q - p_m\|^2 \varphi(q) \mathrm{d}q + \sum_{r=1, i\in\{F,R\}}^{n_b} \int_{\partial V^i_r} \|q - p_r\|^2 \varphi(q) \frac{\partial l}{\partial p_m} \frac{\partial \partial V^i_r}{\partial l} \mathcal{N}_i \mathrm{d}q + \\ \int_{\partial V^F} \|q - p_F\|^2 \varphi(q) \frac{\partial \partial V^F}{\partial p_m} \mathcal{N}_F \mathrm{d}q + \int_{\partial V^R} \|q - p_R\|^2 \varphi(q) \frac{\partial \partial V^R}{\partial p_m} \mathcal{N}_R \mathrm{d}q. \end{aligned} \tag{5.24}$$

Substituting (5.22) into (5.24) and using (5.23), we have

$$\frac{\partial \mathcal{G}}{\partial p_m} = -2\int_{V_m}(q-p_{sm})\varphi(q)\mathrm{d}q + \frac{\partial l}{\partial p_m}\Bigg(\sum_{r=1,i\in\{F,R\}}^{n_b}\Bigg(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l-p_i\|}\int_{\partial V_r^i}\|q-p_r\|^2\varphi(q) \\ (\frac{p_i+l}{2}-q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V_r^i}\|q-p_r\|^2\varphi(q)\mathrm{d}q\Bigg)+ \\ \sum_{i\in\{F,R\}}\Bigg(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l-p_i\|}\int_{\partial V^i}\|q-p_i\|^2\varphi(q)(\frac{p_i+l}{2}-q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V^i}\|q-p_i\|^2\varphi(q)\mathrm{d}q\Bigg)\Bigg). \tag{5.25}$$

Further simplification of (5.25) using equations of mass and centroid (5.2) results in

$$\frac{\partial \mathcal{G}}{\partial p_m} = -2M_{V_m}(C_{V_m}-p_m) + \Bigg(M_{V_m}I_2 + \sum_{p_r\in\{p_m,\mathcal{N}_{p_m}\}}M_{\partial V_r}p_r^\top\Bigg) \\ \Bigg(M_{\mathcal{Q}}I_2 + M_{\partial\mathcal{Q}}l^\top - \sum_{r=1,i\in\{F,R\}}^{n_b}M^i_{\partial V_r}p_r^\top\Bigg)^{-1}\Bigg(\sum_{r=1,i\in\{F,R\}}^{n_b}\Bigg(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l-p_i\|}\int_{\partial V_r^i}\|q-p_r\|^2\varphi(q) \\ (\frac{p_i+l}{2}-q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V_r^i}\|q-p_r\|^2\varphi(q)\mathrm{d}q\Bigg)+ \\ \sum_{i\in\{F,R\}}\Bigg(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l-p_i\|}\int_{\partial V^i}\|q-p_i\|^2\varphi(q)(\frac{p_i+l}{2}-q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V^i}\|q-p_i\|^2\varphi(q)\mathrm{d}q\Bigg)\Bigg). \tag{5.26}$$

As the agents move, the change in their position affects the collective centroid l and the boundaries $\partial\mathcal{Q}$ accordingly. This represents the expected sensitivity of the boundaries of the region $\mathcal{Q}$ on the position of p_m. The integral on the boundaries of $\mathcal{Q}$ is calculated once for all the agents at each time step and is applied to obtain the derivative of the cost function associated with different agents.

5.3.1 Computation of the Voronoi cells

The Voronoi cells associated with each agent may require a different set of information to be computed online. As described before, the agents with edges shared only with other agents are able to compute the Voronoi cell by communicating with the neighboring agents within their group. However, the agents with edges shared with the boundaries ∂Q need the position of the collective centroid of the group, as well as the position of the guidance points p_R and p_F that are known a

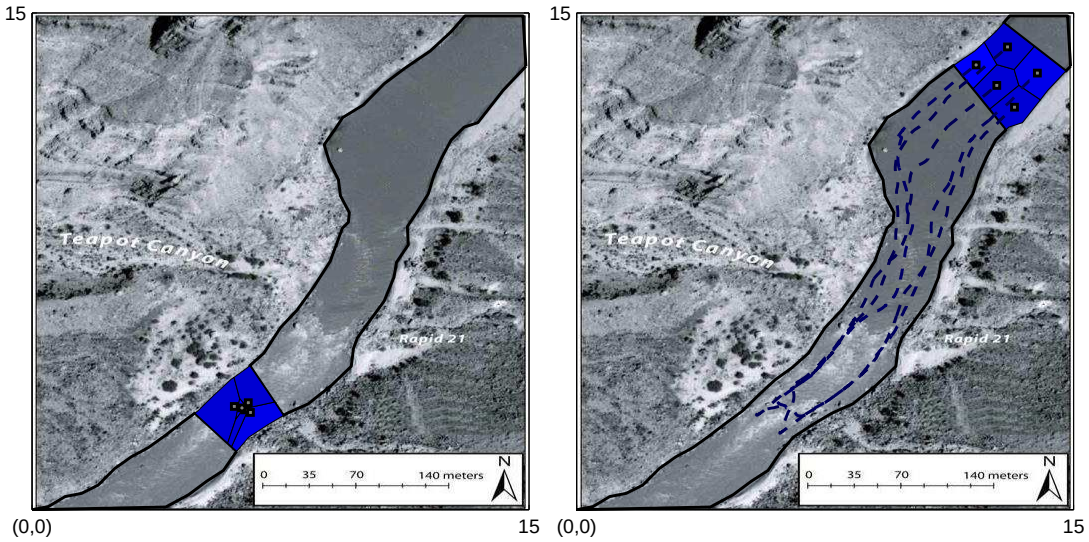


Figure 5.2: Initial (left) and final (right) configurations for a team of five robots maintaining the desired coverage throughout the illustrated section of the Colorado river at Teapot Canyon, the tributary forming rapid 21. Dashed lines depict the traversed trajectory by robots.

priori. We can implement two alternative communication algorithms; first, it is assumed that all agents communicate with each other meaning that each agent has the location of the other agents. Even though this is not an efficient approach, it can be applied for a small number of agents that are in a relatively close neighborhood of other agents due to the nature of the flow problems. These assumptions are realistic due to the fact that the width of the coverage area is limited by the boundaries of the flow and length of the area can be controlled by the relative distance of p_F and p_R.

In an alternative and more efficient communication approach, a greedy algorithm that provides the required data flow within each team and between the so-called leaders of different teams has been proposed in our recent work [3]. To ensure the required data flow, the position of the collective centroid of the group of agents is communicated. This can be implemented when there is a large number of agents covering a large area in some special cases.

5.3.2 CONTROLLER DESIGN

A gradient decent-based control law is proposed here to ensure the optimal coverage in the sense of minimizing the sensing cost function at each time step. The following dynamics is imposed on each agent

$$\dot{p}_m = u_m = \frac{K_m}{2M_{V_m}}\Big(- \frac{\partial \mathcal{G}}{\partial p_m}\Big) = \frac{K_m}{2M_{V_m}}\Big(2M_{V_m}(C_{V_m} - p_m) - \gamma_m\Big), \quad m = 1, \ldots, n, \quad (5.27)$$

where K_m is a positive scalar and

$$\gamma_m = \Big(M_{V_m} I_2 + \sum_{p_r \in \{p_m, \mathcal{N}_{p_m}\}} M_{\partial V_r} p_r^\top\Big)\Big(M_Q I_2 + M_{\partial Q} l^\top - \sum_{r=1, i\in\{F,R\}}^{n_b} M^i_{\partial V_r} p_r^\top\Big)^{-1}$$
$$\Big(\sum_{r=1,i\in\{F,R\}}^{n_b}\Big(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l - p_i\|}\int_{\partial V_r^i}\|q - p_r\|^2\varphi(q)(\frac{p_i + l}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V_r^i}\|q - p_r\|^2\varphi(q)\mathrm{d}q\Big) +$$
$$\sum_{i\in\{F,R\}}\Big(\frac{\mathcal{N}_i(\mathcal{N}_i)^\top - I_2}{\|l - p_i\|}\int_{\partial V^i}\|q - p_i\|^2\varphi(q)(\frac{p_i + l}{2} - q)\mathrm{d}q + \frac{1}{2}\mathcal{N}_i\int_{\partial V^i}\|q - p_i\|^2\varphi(q)\mathrm{d}q\Big)\Big). \quad (5.28)$$

By dividing K_m by the varying term $2M_{V_m}$, the control input is normalized to distribute the effect of both $2M_{V_m}(C_{V_m} - p_m)$ and γ_m in the controller design. While the first term drives the agent towards its centroid, the second term is associated with the changing boundaries of the coverage area $\mathcal{Q}$. In other words, the above control law ensures that the agents are confined within the given dynamic boundaries of $\mathcal{Q}$. Also, as discussed earlier, the guidance points are required to move along the trajectory representing the mean of the boundaries of the river. Hence, the following dynamics are imposed on the guidance points

$$\dot{P}_R = K_R(\partial V^a - P_R), \quad \dot{P}_F = K_F(\partial V^a - P_F), \quad (5.29)$$

where K_F and K_R are positive scalars. Due to the dynamics of the boundaries of the moving coverage area, the imposed dynamics on the guidance points and the agents has to ensure the convergence to the optimal configuration from the coverage perspective. To this end, the time step associated with the varying guidance points that imposes the dynamics on the boundaries and moves the region $\mathcal{Q}$ along the flow field needs to be larger than the time step of the dynamics of the agents. Regardless of the chosen time step, after converging to the first optimal configuration, the

next convergence to new optimal configurations is expected to be faster. It is mainly because each optimal configuration is considered as the initial condition for the next configuration associated with the new Q and hence the location of the agents are near optimal at each time step associated with the guidance points.

5.4 Simulation Results and Discussion

The proposed approach is evaluated using numerical examples that investigate the capability of the proposed method in providing an optimal coverage for flow type problems. As the first example, a section of the river at Teapot Canyon, the tributary forming rapid 21 [86] is chosen to implement the monitoring agents along the main stream of the river. Five robots are deployed to move along the river and gather the required information while maintaining the desired coverage at each portion of the river. As described before, at each iteration the agents are near the optimal location in their respective new coverage area. Hence, as the coverage area moves along the river's streamline, the agents reach their associated optimal position. Finally, as shown in Figure 5.2 the agents converge to a final configuration as the moving coverage area arrives at the final region through the defined trajectory for the guidance points. The dynamics of the agents is obtained through choosing $K_m = 0.21$ and also in this case $K_R = K_F = 0.13$ that allows for a uniform coverage throughout the region.

To evaluate the proposed approach on a different example, a section of Ohio river is used for the coverage purpose as in Figure 5.3. The river is located on the border line of the states of Illinois and Kentucky and is considered to be a wildlife preserve. The given section of the river that is located in a hard-to-access area is chosen to deploy the agents to perform the assigned monitoring task. The results indicate that the agents are capable of moving within the boundaries of the river. The presented results are obtained for $K_m = 0.82$ and we also assign $K_R = K_F = 0.31$. As mentioned before, this approach allows for either expanding or compressing the covered area through changing the relative distance of the guidance points. This can be seen in the Figures 5.4, where the results are shown for both cases. Due to the nature of the coverage task, it is very likely

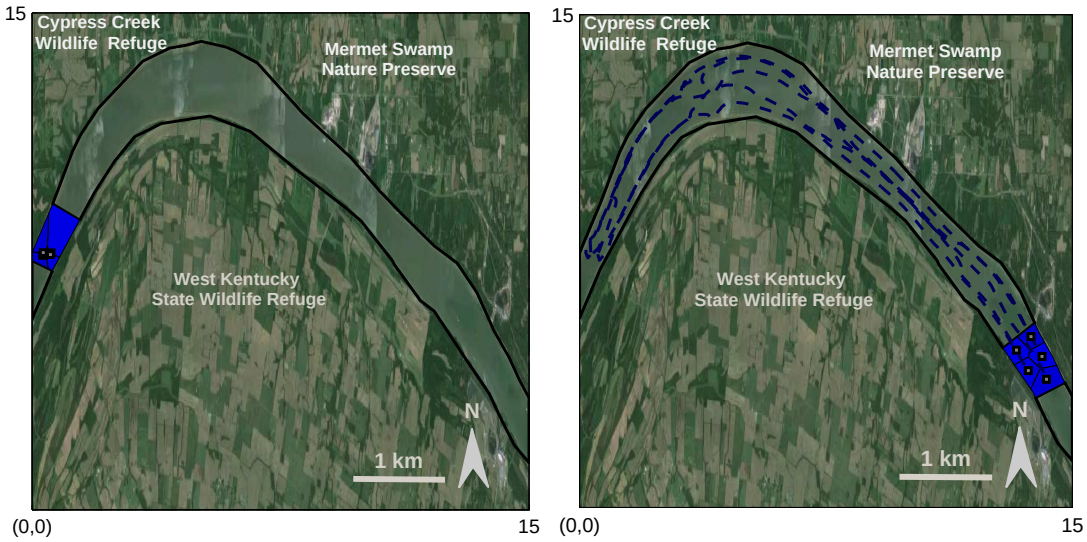


Figure 5.3: Initial (left) and final (right) configurations for a team of five robots maintaining optimum coverage throughout a section of the Ohio river. Dashed lines show the paths of the robots.

that a portion of the river requires a more accurate monitoring. The proposed approach provides a flexible tool by assigning a smaller or larger area to each agent to achieve a stronger or more moderate sensory coverage, respectively. The dynamics on the guidance points are imposed by $K_R = 0.26, K_F = 0.16$ and $K_R = 0.22, K_F = 0.33$ for the compression and expansion examples, respectively.

5.5 Concluding Remarks

To cope with the complexity of the coverage problems in real world applications, a coverage approach for flow fields is presented in this paper. The proposed approach ensures the optimum coverage over the flow surface in a local way leading to the partitioning of the dynamic region by dividing it into multiple subregions associated with multiple deployed agents. This is beneficial to handle tasks where it is required to cover a long environment like rivers. Hence, as the coverage

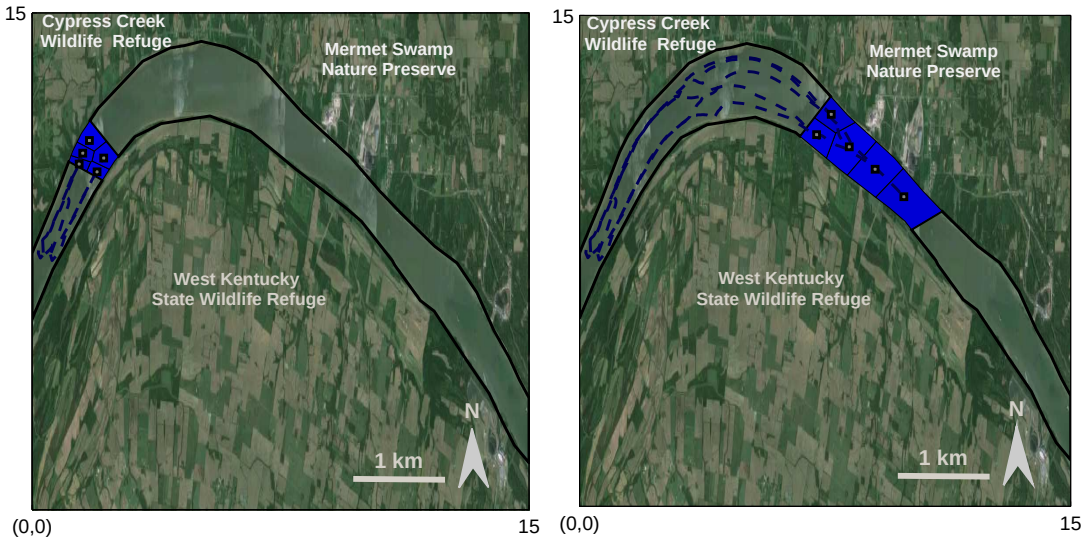


Figure 5.4: Compressed (left) and expanded (right) configurations for a team of five robots maintaining an optimum coverage. Dashed lines show the paths of the robots.

area moves within the boundaries of the desired environment, the agents provide the optimal coverage for each dynamic region. The developed scheme also allows for compressing or expanding the coverage area. This is particularly useful when it is required to gather more accurate data in a section of the river or a larger area needs to be covered at that specific section of the river. In essence, the proposed approach provides the means to monitor long rivers located in remote areas via the proposed control laws that can guarantee the optimum coverage.

CHAPTER 6

A SUPPORT VECTOR MACHINE-BASED METHOD FOR LPV-ARX IDENTIFICATION WITH NOISY SCHEDULING PARAMETERS [1]

[1] F. Abbasi, J. Mohammadpour, R. Tóth, and N. Meskin. A Support Vector Machine-based Method for LPV-ARX Identification with Noisy Scheduling Parameters. 2014. In *Proc. 13*th *European Control Conference*, Strasbourg, France: pp. 2744-2749.

ABSTRACT

In this paper, we present a method that utilizes support vector machines (SVM) to identify linear parameter-varying (LPV) auto-regressive exogenous input (ARX) models corrupted by not only noise, but also uncertainties in the LPV scheduling variables. The proposed method employs SVM and takes advantage of the so-called "kernel trick" to allow for the identification of the LPV-ARX model structure solely based on the input-output data. The objective function, as defined in this paper, allows to consider uncertainties related to the LPV scheduling parameters, and hence results in a new formulation that provides a more accurate estimation of the LPV model in the presence of scheduling uncertainties. We further demonstrate the viability of the proposed LPV identification method through numerical examples, where we show that higher best fit rate (BFR) can be achieved under realistic noise conditions using the proposed method compared to the method initially proposed in [146].

6.1 INTRODUCTION

Identification of linear parameter-varying (LPV) systems has attracted the attention of many researchers within the control systems community (see [145] and many references therein). The basic idea in identifying an LPV model is to introduce a parametrization of the underlying dependency of the model on the scheduling variables in terms of *a priori* chosen set of basis functions. The very first works on LPV system identification assumed a prior knowledge of the basis functions and focused on the identification of the unknown parameters [13, 138]. In those early works, the problem of finding unknown parameters was simply formulated as a least-squares (LS) problem. Making the assumption that the model structure is known is sometimes valid since the LPV model can be derived directly from the nonlinear system equations; however, this is not always the case and hence additional efforts must be devoted to identify the basis functions. Also, inaccurate selection of the basis functions leads to structural bias while over-parametrization

results in a variance increase of the estimates. To resolve the high computational load and bias-variance trade-off arising from over-parameterizations based techniques for least-squares based model estimation, a semi-parametric identification approach based on least-squares support vector machines (LS-SVM) was introduced for a class of nonlinear regression models [27, 48, 74]. Some recent works have been done to address a similar problem for LPV model identification using LS-SVM [75, 146].

Support vector machines are supervised learning tools originated in modern statistical learning theory that can effectively provide a non-parametric estimation of the dependency structure for linear regression based LPV models [91, 147]. The supervised learning method was originally proposed by [131, 147] to rebuild the inherent functional relationships and structures in the data [22]. This non-parametric functional dependence estimation is more successful in coping with the bias-variance trade-off than semi-parametric approaches like dispersion functions methods [146]. Also, considering l_2 loss functions in the LS-SVM approach gives a variation of the original SVM method that presents an effective model structure learning in the LPV setting. Finding computationally efficient and unique solution of the linear problem are the advantages of these slightly different approaches like LS-SVM over original SVM method. Hence our aim in this work is to employ an effective variation of the LS-SVM method combined with a cost function that focuses not only on prediction error, but also weighs possible uncertainties in the system variables.

Accurate knowledge of scheduling signals is a critical assumption in both LPV system identification and LPV control design. The previous works [22, 75, 146] that use the kernel-based SVM for "model learning" assumes the perfect knowledge of the scheduling signal during the system identification process. The questions that we address in this work are: (i) how is the performance of the LPV system identification procedure proposed in [146] affected in the presence of such uncertainties? and (ii) how can we improve the LPV system identification when such uncertainties exist? We will examine the first question through simulation studies. Also, to address the latter question, we model such uncertainties in LPV parameters (that we refer to as "error in variables") and include them in the cost function associated with the underlying optimization problem. In conjunction with

SVM, the proposed objective function finds the LPV model structure and the corresponding model coefficients in the presence of error in the variables. This is done using the so-called *kernel trick* approach instead of explicitly defining the feature maps (*i.e.*, basis functions) involved [146].

The rest of this paper is organized as follows: Section II describes the basic formulation for the LPV model identification problem studied here. Sections III presents the proposed identification method (that we refer to as EIV-SVM). Simulation results are shown in Section IV and finally, concluding remarks will be made.

6.2 Identification of LPV Input/Output Models

We assume that the following SISO deffrence equation defines the behaviour of the data generating system,

$$y(t) = \sum_{i=1}^{n_a} a_i(p(t))y(t-i) + \sum_{j=0}^{n_b} b_j(p(t))u(t-j) + e(t), \tag{6.1}$$

where t represents the discrete time, y and u are the outputs and inputs of the system, and e represents a white stochastic noise process. We further assume that the coefficients a_i and b_j are dependent on the time-varying scheduling variable(s) $p(t)$. Note that (6.1) defines an auto regressive with exogenous input (ARX) dynamic structure. For identification of system (6.1), we will adapt the same model structure where the orders of n_a and n_b are assumed to be known. Commonly, in the LPV system identification, when the number of coefficient functions a_i and b_j is decided, then the dependence of the coefficients on $p(t)$ is parameterized as a linear combination of a finite number of basis functions with static dependence on p chosen *a priori*

$$a_i(p(t)) = \sum_{r=1}^{n_g} \alpha_{i,r}\psi_{i,r}(p(t)) \qquad i = 1, ..., n_a$$
$$b_j(p(t)) = \sum_{r=1}^{n_g} \beta_{j,r}\psi_{j,r}(p(t)) \qquad j = 0, ..., n_b,$$

where $\{\psi_{i,r}\}_{i=1,r=1}^{n_a,n_g}$ and $\{\psi_{j,r}\}_{j=0,r=1}^{n_b,n_g}$ are basis functions of the system coefficients. As described earlier in the paper, since improper selection of basis functions can cause structural bias, best choice of these functions is crucial. Our aim in this paper is to employ the so-called *kernel trick*

in order to avoid the difficulties arising from choosing basis functions in a non-systematic way. As described later in the paper, tuning the kernel function parameters has a significant impact on the accuracy of the identified LPV model. In fact, the bias-variance trade-off is tuned, which means achieving a higher accuracy by tuning the parameters causes more sensitivity to noise. We next describe all the coefficients and basis functions in a compact LPV-ARX form and put them in a matrix form. To do so, we first define $x(t)$ as an $n_g = n_a + n_b + 1$ dimensional vector containing all the outputs and inputs as

$$x(t) = \begin{bmatrix} y(t-1) & \dots & y(t-n_a) & u(t) & \dots & u(t-n_b) \end{bmatrix}^\top,$$

and

$$\begin{bmatrix} a_1 & \dots & a_{n_a} & b_0 & \dots & b_{n_b} \end{bmatrix} = \\ \begin{bmatrix} \rho_1^\top \phi_1(p(t)) & \dots & \rho_{n_g}^\top \phi_{n_g}(p(t)) \end{bmatrix},$$

where $\phi_i(p(t))$ is a nonlinear vector map from the scheduling signal space $\mathbb{P}$ to an n_H-dimensional space. ρ_i is a parameter in $\mathbb{R}^{n_H}$. Theoretically, n_H can be infinite, except in parametric LPV identification, where the number of basis functions is set *a priori*. Employing the aforementioned setup, the LPV-ARX model of (6.1) can be written in a compact form as

$$y(t) = \begin{bmatrix} \rho_1^\top \phi_1(p(t)) & \dots & \rho_{n_g}^\top \phi_{n_g}(p(t)) \end{bmatrix} x(t) + e(t),$$

or

$$y(t) = \rho^\top \Phi + e(t), \tag{6.2}$$

where

$$\Phi = \begin{bmatrix} \phi_1(p(t))x_1(t) & \dots & \phi_{n_g}(p(t))x_{n_g}(t) \end{bmatrix}^\top. \tag{6.3}$$

6.3 LPV Model Identification Using LS-SVM

Least-squares (LS)-based algorithms have been widely utilized for system identification of linear and nonlinear systems in a regression form [82]. In addition, they have been applied for LPV I/O

model identification with linear predictors using *a priori* specified parametrization of the dependencies [29, 35]. With the use of LS-SVM for LPV model identification, first proposed in [146], the dependence of the basis functions on the LPV parameters is assumed to be unspecified. The idea behind the work by Tóth *et al.* [75, 146] is that the time-varying coefficients of the LPV model described in an input/output form can be estimated using the so-called *kernel trick* method without assigning specific basis functions. In fact, inherent nonlinearity of the coefficient dependencies can be "learned" efficiently in a projected high-dimensional feature space [146].

6.3.1 An LS-SVM Estimator under Uncertain/Noisy Scheduling

In this paper, we extend the work in [75, 146] to develop an SVM-based identification method that can cope with observation/measurement errors in the scheduling variable $p(t)$. To this purpose, we represent the LPV model in a regression form that is appropriate within the SVM setting, as follows

$$y(t) = \sum_{i=1}^{n_g} [\rho_i^\top \phi_i(p(t)) + \Delta v_i(p(t))] x_i(t) + e(t), \tag{6.4}$$

where Δv_i represents the uncertainties in the i^{th} coefficient function caused by errors by, *e.g.*, the measurement process. The purity ratio of distillation columns that is used as the scheduling parameter in the LPV identification of the process, is an example of roughly measured scheduling variables that always contain some observation/measurement error. We note that Δv_i is naturally different than the environmental noise $e(t)$ that is directly added to the system output. The error-in-variable terms Δv_i, captured in (6.4), can negatively affect the LPV system identification since the data collected from the system, *i.e.*, $x_i(t)$, are based on noise-free scheduling trajectory actually influencing the system, while the measured scheduling trajectory obtained for the model identification purposes is noisy. To model the impact of error in variables in the SVM formulation, we add these uncertainties directly to the coefficient functions to be identified as

$$y(t) = [a_1 + \Delta a_1 \ \dots \ a_{n_a} + \Delta a_{n_a} \ \ b_0 + \Delta b_0 \ \dots \ b_{n_b} + \Delta b_{n_b}] x(t) + e(t).$$

Using the basis function formulation of the model coefficients, we have

$$y(t) = [\rho_1^\top \phi_1(p(t)) + \Delta v_1 \ \ldots \ \rho_{n_g}^\top \phi_{n_g}(p(t)) + \Delta v_{n_g}]x(t) + e(t) \tag{6.5}$$

or

$$y(t) = \rho^\top \Phi + \Delta V^\top x(t) + e(t),$$

where Φ was defined by (6.3) and the error in variables are lumped into a vector ΔV defined by

$$\Delta V^\top = \begin{bmatrix} \Delta v_1 & \ldots & \Delta v_{n_g} \end{bmatrix}.$$

Note that ΔV is considered to be stochastic with $\mathbb{E}\{\Delta V\} = 0$.

6.3.2 SVM Regression with Error in Variables

To characterize an estimate for the model presented in (6.4), we propose the following cost function

$$J(\rho, e, \Delta V) = \frac{\gamma}{2} \left\| \begin{bmatrix} \Delta V & e \end{bmatrix} \right\|_F + \frac{1}{2} \sum_{i=1}^{n_g} \rho_i^\top \rho_i, \tag{6.6}$$

which is inspired by the standard cost function used in the total least-squares (TLS) method, that can cope with both error-in-variables and measurement noise [79, 88, 121]. In the cost function above, γ is the regularization parameter. We then expand the matrices and the Frobenius norm and assign different weights (regularization parameters) to ΔV and e resulting in

$$J(\rho, e, \Delta V) = \frac{\gamma_1}{2} \sum_{t=1}^{N} \sum_{i=1}^{n_g} \Delta v_i^\top(t) \Delta v_i(t) + \frac{\gamma_2}{2} \sum_{t=1}^{N} e^2(t) + \frac{1}{2} \sum_{i=1}^{n_g} \rho_i^\top \rho_i. \tag{6.7}$$

where γ_1 and γ_2 expresses the trade-off between te l_2-loss (prediction error) and l_2-coefficient deviation and regularization in this multi-objective cost function. Note that Δv_i decouples from $e(t)$ due to its correlation with γ in (6.6).

6.3.3 Constrained Optimization Problem

The optimization problem described earlier in this section is solved using the Lagrangian method considering the LPV model in the regression form as the problem constraint. The overall objective

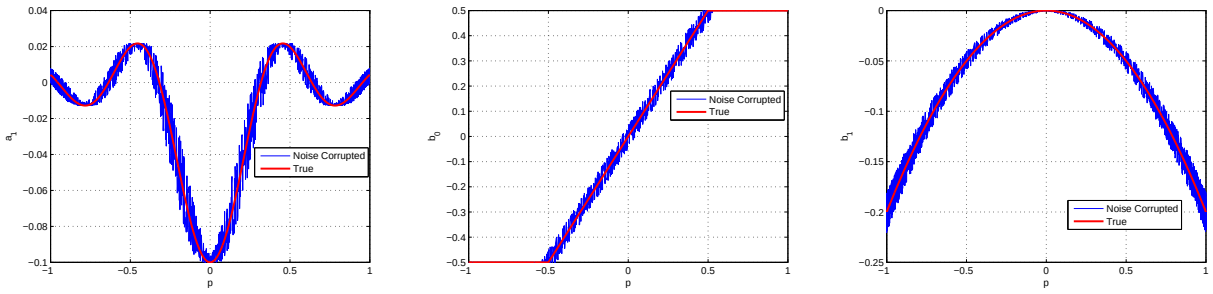


Figure 6.1: LPV system coefficients corrupted by noise

is now to solve the following problem

$$\min_{\{\rho,e,\Delta V\}} J(\rho,e,\Delta V) = \frac{\gamma_1}{2}\sum_{t=1}^{N}\sum_{i=1}^{n_g}\Delta v_i^\top(t)\Delta v_i(t) + \frac{\gamma_2}{2}\sum_{t=1}^{N}e^2(t) + \frac{1}{2}\sum_{i=1}^{n_g}\rho_i^\top\rho_i$$

$$s.t.\;\; y(t) = \sum_{i=1}^{n_g}[\rho_i^\top\phi_i(p(t)) + \Delta v_i(p(t))]x_i(t) + e(t).$$

The error function variables can be determined by setting the Lagrangian for this constrained optimization problem as

$$L(\rho,e,\Delta V,\alpha) = J(\rho,e,\Delta V) - \big[\sum_{t=1}^{N}\alpha_t\sum_{i=1}^{n_g}[\rho_i^\top\phi_i(p(t)) + \Delta v_i(p(t))]x_i(t) + e(t) - y(t)\big],$$

where α_t's are the Lagrangian multipliers. We then employ the Karush-Kuhn-Tucker (KKT) condition to find the saddle point of L which under the zero-duality gap corresponds also to the optimum of J,

$$\begin{aligned}
\frac{\partial L}{\partial \Delta v_i} = 0 \;\;&\rightarrow\;\; \Delta v_i(t) = \frac{\alpha_t}{\gamma_1}x_i(t)\\
\frac{\partial L}{\partial e(t)} = 0 \;\;&\rightarrow\;\; e_i(t) = \frac{\alpha_t}{\gamma_2}\\
\frac{\partial L}{\partial \rho_i} = 0 \;\;&\rightarrow\;\; \rho_i = \sum_{i=1}^{n_g}\alpha_t\phi_i(t)x_i(t)\\
\frac{\partial L}{\partial \alpha_t} = 0 \;\;&\rightarrow\;\; e(t) = y(t) - \sum_{i=1}^{n_g}[\rho_i^\top\phi_i(t) + \Delta v_i(t)]x_i(t).
\end{aligned} \tag{6.8}$$

Substituting the obtained variables back into (6.8) results in

$$y(t) = \sum_{i=1}^{n_g} \big\{ [\sum_{j=1}^{N} \alpha_j x_i(j)\phi_i^\top(j)]\phi_i(t) + \alpha_t \gamma_1^{-1} x_i(t) \big\} x_i(t) + \gamma_2^{-1}\alpha_t.$$

By collecting the related terms together, we have

$$y(t) = \sum_{i=1}^{n_g} \sum_{t=1}^{N} \alpha_j \underbrace{x_i(j)\phi_i^\top(j)\phi_i(t)x_i(t)}_{[\Omega]_{j,t}^i} + \gamma_1^{-1}\alpha_t x_i(t)x_i(t) + \gamma_2^{-1}\alpha_t, \tag{6.9}$$

where we then define

$$[\Omega]_{j,t} = \sum_{i=1}^{n_g} [\Omega]_{j,t}^i = \sum_{i=1}^{n_g} x_i(j)\phi_i^\top(j)\phi_i(t)x_i(t). \tag{6.10}$$

that can allow us to write (6.9) in the matrix form considering the discrete time instants $t = 1, \ldots, N$. This leads to the following expression

$$Y = (\Omega + \gamma_1^{-1}\text{diag}(\sum_{i=1}^{n_g} x_i^2(1),\ \ldots\ , \sum_{i=1}^{n_g} x_i^2(N)) + \gamma_2^{-1} I_N)\alpha. \tag{6.11}$$

Writing the first term of (6.11) in the kernel form as in [146] yields a systematic way to cope with the basis functions complexity. In fact, this new formulation is based on the *kernel trick* that estimates the inner product of the feature maps in a lower dimensional space without any need to directly define these functions. The elements of the matrix Ω are defined by

$$\begin{aligned} [\Omega]_{j,t}^i &= x_i(j)\phi_i^\top(j)\phi_i(t)x_i(t) \\ &= x_i(j)\langle \phi_i^\top(j), \phi_i(t)\rangle x_i(t) \\ &= x_i(j)(K^i(p(j), p(t)))x_i(t). \end{aligned}$$

where K^i is a positive definite kernel function that satisfies Mercer's conditions in the inner product $\langle \phi_i(j), \phi_i(t)\rangle$ space without explicitly calculating the mapping. In fact, the *kernel trick* only requires the calculation of the modified inner product using every pair of data points and the kernel function's value instead of knowing the basis functions. Although, choosing the most appropriate kernel highly depends on the problem at hand and fine tuning of its parameters can easily become a tedious and cumbersome task, the choice of a particular kernel can be very intuitive

and straightforward depending on what kind of information we are expecting to extract from the data. Among various possible choices for kernel functions, the use of *radial basis function (RBF)*, *polynomial*, and *sigmoid function* is appealing due to their ability to represent the nonlinearities in different types of data [131]. In this paper, we use the above three kernel functions and their performance is compared in the next section. The following equation represents the RBF kernel function

$$K^i(p(j), p(t)) = \exp\big(-\frac{\|p(j) - p(t)\|_2^2}{2\sigma_i^2}\big), \tag{6.12}$$

where σ_i is an adjustable parameter. The *polynomial* kernels are represented by

$$K^i(p(j), p(t)) = \big(1 + \frac{p(t)^\top p(j)}{c}\big)^d, \tag{6.13}$$

where adjustable parameters are the slope *c* and the polynomial degree *d*. Finally, the implemented *sigmoid* kernel function is

$$K^i(p(j), p(t)) = \tanh\big(\lambda p(t)^\top p(j) + \beta\big), \tag{6.14}$$

where λ and β are the tuning parameters.

After substitution of the chosen kernel function into Eq. (6.11), the solution to this linear equation is given by

$$\alpha = (\Omega + \gamma_1^{-1}\mathrm{diag}(\sum_{i=1}^{n_g} x_i^2(1),\ \dots\ , \sum_{i=1}^{n_g} x_i^2(N)) + \gamma_2^{-1} I_N)^{-1} Y.$$

Using the obtained expression for α and the *kernel trick* approach, coefficients of the LPV-ARX model estimate are calculated as

$$a_i(\cdot) = \rho_i^\top \phi_i(\cdot) + \Delta v_i(t) = \sum_{t=1}^{N} \alpha_t x_i(t) K^i(p(t), \cdot) + \frac{\alpha_t}{\gamma_1} x_i(t),$$
$$b_j(\cdot) = \rho_j^\top \phi_j(\cdot) + \Delta v_j(t) = \sum_{t=1}^{N} \alpha_t x_j(t) K^j(p(t), \cdot) + \frac{\alpha_t}{\gamma_1} x_j(t).$$

where N is the number of the measurements.

6.4 SIMULATION RESULTS

In order to evaluate the efficiency of the proposed LS-SVM-based LPV model identification method (that we hereby refer to as "EIV-SVM") when the LPV parameters are corrupted by noise, we apply it to the example in [146]. The following LPV model, in an input/output form is considered:

$$y(t) = a_1(p(t))y(t-1)) + \sum_{i=0}^{1} b_i(p(t))u(t-i) + e_0(t), \tag{6.15}$$

where $p(t) \in [-1\ 1]$. To generate data for identifying the system described by (6.15), $N = 1500$ samples of data points have been simulated using $u(t) = \sin(\frac{\pi}{2}t)$, $p(t) = \sin(\frac{t}{4})$ and independent and identically distributed (i.i.d.) e_0 with $e_0 \sim \mathcal{U}(-1, 1)$. We also assume that instead of $p(t)$ only $p^*(t) = p(t) + w(t)$ is available to be measured in the system where w is also i.i.d. and $w(t) \sim \eta \times \mathcal{U}(-1, 1)$ where η is a coefficient to control the noise level in the scheduling variable. In the following example, η is assigned 0.05 and 0.1 for the first and second cases, respectively. It should be noted that to avoid clipping of the distribution of η here $p^*(t)$ is allowed to deviate from $[-1\ 1]$.

Table 6.1: The MSE and BFR of the EIV-SVM and LS-based SVM methods over 100 runs

		LS-SVM (RBF)		EIV-SVM (RBF)		EIV-SVM (polynomial)		EIV-SVM (sigmoid)	
Noise Level		MSE	BFR	MSE	BFR	MSE	BFR	MSE	BFR
Case I	Mean	5.4905e-04	0.8326	4.9695e-04	0.8867	5.4534e-04	0.8785	6.1577e-04	0.8723
	Std	7.4846e-06	0.0031	6.9000e-06	0.0032	6.5217e-06	0.0035	7.1842e-6	0.0042
Case II	Mean	7.7970e-04	0.7660	6.2417e-04	0.8128	6.6442e-4	0.8046	7.1246e-4	0.8026
	Std	7.3813e-05	0.0068	5.2667e-05	0.0056	6.3196e-4	0.0064	5.9834e-6	0.0062

The coefficients of the LPV system above are considered to have the following nonlinear dependencies on the scheduling variable

$$b_0(p(t)) = \begin{cases} +0.5 & \text{if } p(t) > 0.5 \\ p(t) & \text{if } -0.5 < p(t) < 0.5 \\ -0.5 & \text{if } p(t) < -0.5 \end{cases}$$

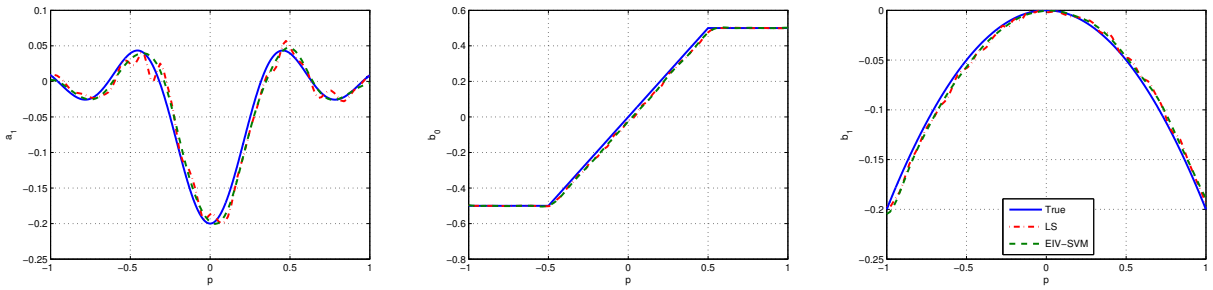


Figure 6.2: Comparison of LPV model identification using the proposed method in this paper and that in [146]: LS and EIV-SVM, respectively, represent the LS-SVM based method in [146], and the LS-SVM based method proposed in this paper to cope with the error in variables.

$$b_1(p(t)) = -0.2 \times p_t^2$$
$$a_1(p(t)) = -0.1 \times \frac{\sin(\pi^2 p(t))}{\pi^2 p(t)}.$$

We illustrate two sets of simulation results. First, we compare the accuracy of the LPV model identification approach in this paper with that in [146] considering an RBF kernel function for both cases. As described earlier, in addition to a white Gaussian noise added to the system output with signal to noise ratio of 30dB, another white Gaussian noise is directly added to the scheduling parameter that affects the three LPV model coefficients, as depicted in Figure 1. The results of one run of simulations using the noisy scheduling parameter $p^*(t)$ are illustrated in Figure 2. As observed from the three subplots, the proposed method in this paper outperforms the LS-SVM method in [146] in identifying the three parameter-varying coefficients b_0, b_1 and a_1. It is noted that the hyperparameters γ_1, γ_2, used for the model learning have been tuned with a trial-and-error.

In the second set of simulation results, we compare the three kernel functions described in the previous section to evaluate the performance of the proposed SVM-based model identification approach in the presence of error in variable. To examine the accuracy of the proposed identification method and compare it with the previous work of Tóth *et al.* [146], we consider two error

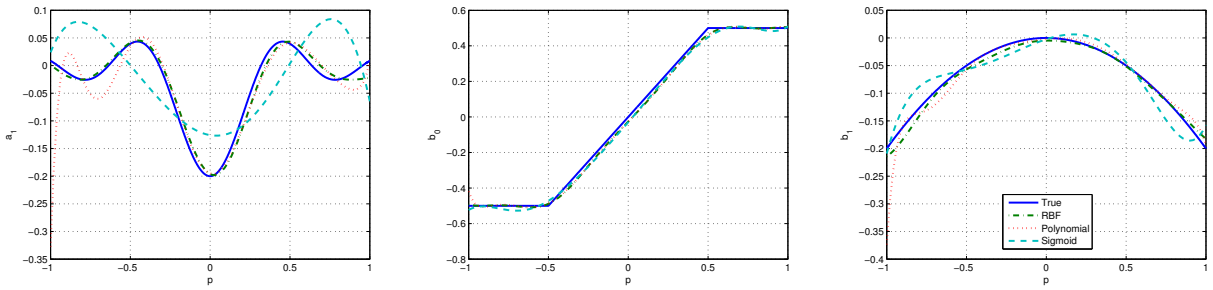


Figure 6.3: Estimated LPV model coefficients using RBF, polynomial, and sigmoid kernels: The results illustrate that for the data generated from the given LPV system, the RBF kernel outperforms the other two kernel functions in terms of accurately calculating the coefficient functions.

measures of mean square error (MSE) and best fit rate (BFR) defined as

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t))^2,$$

$$\text{BFR} = \max\{0, 1 - \frac{\| y(t) - \hat{y}(t) \|_2}{\| y(t) - \bar{y} \|_2}\},$$

where $\bar{y}$ is the mean of the output in the validation data set, $y(t)$, and $\hat{y}(t)$ is the simulated output. Similar to the first simulation, measurements are corrupted by white Gaussian noise and also a white Gaussian noise directly added to the scheduling variable that affects the three LPV model coefficients (as shown in Figure 1). The comparative analysis is done for two different noise levels added to the scheduling parameter $p(t)$. In the two cases examined, noise signals are generated by $\mathcal{U}(-1, 1)$ multiplied by 0.05 (case I) and 0.1 (case II), respectively. A Monte-Carlo simulation study is performed for a numerical illustration of the identification algorithms through changing random white Gaussian noise in the scheduling variable. In addition, we employ three kernels described in the previous section to evaluate the performance of the proposed SVM-based model identification approach. The regularization parameters are selected through trial and error as $\gamma_1 = 1200$ and $\gamma_2 = 6000$. Also, the parameters associated with each one of the three kernel functions were tuned by cross-validation.

The results of 100 runs are analyzed and the mean and standard variations of the BFR and MSE values are shown in Table 6.1 indicating that the proposed EIV-SVM method of this paper leads to a better approximation of the LPV model coefficients. In addition, the subplots in Figure 3 illustrate the estimates of the three LPV model coefficients as a function of the LPV parameter $p(t)$ for three kernel functions with the proposed EIV-SVM method. We note that the same error in variable approach, as in the first set of simulations, is considered here. Also, the presented results in Table 6.1 and Figure 3 indicate that the RBF kernel (with the tuned parameters $\sigma_1 = \sigma_2 = \sigma_3 = 0.5$) outperforms the other two kernels due to its capability to characterize nonlinearities in the collected data from the LPV model.

To summarize the simulation results, the plots demonstrate that, in the presence of noise in the scheduling variables, the proposed EIV-SVM method exhibits an improved capability of identify the structure of the coefficient functions compared to the LS-SVM method proposed first in [146]. The Monte-Carlo simulation results also showed that the proposed EIV-SVM method not only increases the BFR of the estimated output, but also lowers the standard deviation.

6.5 Concluding Remarks

We presented in this paper new results on the extension of LS-SVM as a powerful machine learning tool for model identification of LPV systems in input/output form. The problem was formulated in a way to yield a solution that can handle errors in the scheduling variables. The cost function we defined in the SVM setting included an additional term associated with the errors in variables. This allowed the kernel-based identification method to partially compensate for the error in p to avoid misestimating of the system parameters and lead to a set of new expressions (compared to [146]) for LPV model coefficients by changing the basis functions.

CHAPTER 7

A BAYESIAN APPROACH FOR MODEL IDENTIFICATION OF LPV SYSTEMS WITH UNCERTAIN SCHEDULING VARIABLES [1]

[1] F. Abbasi, J. Mohammadpour, R. Tóth, and N. Meskin: A Bayesian Approach for Model Identification of LPV Systems with Uncertain Scheduling Variables. 2015. In *Proc. of 54th IEEE Conference on Decision and Control*, Osaka, Japan: pp. 789-794.

ABSTRACT

This paper presents a Gaussian Process (GP) based Bayesian method that takes into account the effect of additive noise on the scheduling variables for identification of linear parameter-varying (LPV) models in input-output representation form. The proposed method approximates the noise-free coefficient functions by a local linear expansion on the observed scheduling variables. Therefore, additive noise on the scheduling variables is reconstructed as a corrective term added on the output noise that is proportional to the squared gradient obtained from the posterior of the Gaussian Process. An iterative procedure is given such that the obtained solution converges to the best estimation of the coefficient functions according to the given measure of fitness. Moreover, the expectation and covariance functions estimated by GP are modified for the noisy scheduling variable case to include noise contribution on the estimated expectation and covariance functions. The model training procedure identifies noise level in the measurements including outputs and scheduling variables by estimating the noise variances, as well as the other defined hyperparameters. Then, the identified distribution of noise signals on the scheduling variables is utilized to denoise the associated scheduling variable signals using a time-domain kernel function. Finally, the performance of the proposed method is compared to the standard GP through two numerical examples.

7.1 INTRODUCTION

Most of the existing methods for model identification of *linear parameter-varying* (LPV) systems consider the scheduling variables to be noise free. However, the presence of uncertainty, i.e., noise, in the measured data including the scheduling variables is inevitable and can lead to an inaccurate model identification. Hence, the precise knowledge of scheduling variables in the presence of uncertainties is a critical issue in both LPV model identification and LPV control design.

Several identification methods have been recently proposed to cope with noisy scheduling variables corresponding to the so-called error-in-variables problem in the context of *linear time-*

invariant (LTI) systems [61, 132, 154]. Unlike the LTI framework, nonlinear dependency of the LPV model coefficients on the scheduling variables is considered to be the main source of complexity in coping with the noise corrupted scheduling variables. There are very few works examining the model identification of LPV systems considering noise corrupted scheduling variables. The previous works [23–25] have focused on the identification of LPV *input/output* (LPV-IO) models using set-membership and *instrumental variable* (IV) based methods. More specifically, a convex relaxation approach is proposed in [25] under the assumption that all the noisy observations including outputs and scheduling variables are bounded. Moreover, the IV-based method presented in [23] is capable of coping with noisy scheduling variables assuming that the instrument is uncorrelated with the scheduling variable noise and the scheduling dependency is linear. More recently, a bias-corrected, IV-based method has been developed for the identification of LPV models from noise corrupted measurements of the outputs and the scheduling variables [113]. While, the recent works have offered significant improvement for the identification of LPV systems, they, however, have assumed that the dependency on the scheduling variables is *a priori* known. The present work introduces a Bayesian-based approach assuming *a priori* unknown dependency, characterized in terms of prior distribution, on the noise corrupted scheduling variables. The Bayesian-based approaches provide a rich variety of *a priori* kernels that can effectively characterize such distributions and hence identify structural characteristics of the systems under study [30, 114, 115].

The Bayesian formulation is based on the expression of the beliefs about the prior information or measurements through specification of *a priori* knowledge before observing new data. The predictions are made by averaging over all possible predictive distributions while they are weighted by their respective posterior probability. Unlike the Bayesian methods, non-Bayesian schemes select a specific parameter or data over others by some predefined criterion [119]. In addition, according to the Bayesian inference as an effective statistical inference, the probability of a hypothesis is updated as a new evidence is captured.

Gaussian process (GP) models generalize the Gaussian probability distribution to the function space that is essential for black-box regression problems. They can be seen as the Bayesian

version of the well-known support vector machines (SVMs) that provide probabilistic approaches to learning with reproducing kernel Hilbert spaces. Nonparametric Gaussian process models have been widely used in model identification of nonlinear dynamic systems. The predictive performance of GPs has been evaluated in [118] and compared to other modeling approaches like neural networks or local learning methods. In [47], a k-step ahead forecasting of a discrete-time nonlinear and LTI dynamic system is performed using repeated one-step ahead predictions. In the LPV system identification framework, the authors have introduced a Bayesian framework for identification of the coefficients in *finite impulse response* (FIR) dynamic structures in [49]. In this paper, an extension of the standard GP method is formulated to identify the dependency of the LPV model coefficients on the scheduling variables while they are corrupted with a Gaussian noise process.

Throughout this paper, notation $A \odot B$ is used to represent the Hadamard product of the matrices A and B of the same dimension such that $[A \odot B]_{ij} = [A_{ij}] \cdot [B_{ij}]$. In addition, I_N, $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{R}^n$ denote the $N \times N$ identity matrix, the set of real numbers, the set of integer numbers and the set of n-dimensional vector space with real elements, respectively, and $(.)^\top$ represents the transpose of the associated vector or matrix.

The rest of the paper is organized as follows. Section II describes the LPV model formulation. Section III explains the principles of the Gaussian processes. The formulation of the error in the scheduling variables problem is given in Section IV. Section V provides the proposed two-step estimation procedure for LPV model identification. Finally, simulation results are shown in Section VI, and concluding remarks are provided in Section VII.

7.2 LPV Input-Output Models

We consider a *single-input single-output* (SISO) linear parameter-varying (LPV) system in the auto-regressive form with exogenous input (ARX) described by

$$y(k) = -\sum_{i=1}^{n_a} a_i(p(k))y(k-i) + \sum_{i=0}^{n_b} b_i(p(k))u(k-i) + e(k), \tag{7.1}$$

where $k \in \mathbb{Z}$ denotes the discrete time, $y : \mathbb{Z} \to \mathbb{R}$ is the system output and $u : \mathbb{Z} \to \mathbb{R}$ is the system input. Also, $p : \mathbb{Z} \to \mathbb{P}$ is the so-called *scheduling variable* with $\mathbb{P} \subseteq \mathbb{R}^{n_p}$ and

$e(k)$ is an *independent and identically distributed* (i.i.d.) white stochastic noise process that is independent of u and p. The coefficients $a_i(p(k)), b_i(p(k))$ are assumed to be bounded possibly nonlinear functions over $\mathbb{P}$ that fully characterize the LPV model (7.1). To estimate the structure of the model coefficient functions usually requires the parametrization of a_i and b_i in terms of *a priori* known basis functions. To avoid difficulties arising from an inappropriate selection of the basis functions, nonparametric approaches have been proposed in the literature [131]. This can favor the LPV modeling of time-varying or nonlinear systems specially in case of the dynamic dependencies of the model coefficients on the scheduling variables, i.e., dependency on $p(k),\ p(k-1), \ldots$.

Both parametric and nonparametric approaches for LPV model identification aim at describing the underlying dependencies of the model coefficients on the scheduling variables. However, often only a measured version of $p(k)$ is available, polluted by noise, that the dependencies of the coefficient functions on the scheduling variables often leads to bias of the estimated coefficient functions, referred to as an *error-in-variables* problem. The present work is an effort to identify and compensate for the error in the scheduling variables by a modified Bayesian approach. The model (7.1) can be represented in a more compact form by introducing the following notations:

$$x_i(k) = -y(k-i),\ i = 1, \ldots, n_a, \tag{7.2}$$

$$x_{n_a+j+1}(k) = u(k-j),\ j = 0, \ldots, n_b, \tag{7.3}$$

$$\mathbf{x}(k) = [x_1(k)\ x_2(k)\ \ldots\ x_{n_a+n_b+1}(k)]^\top. \tag{7.4}$$

Additionally, the coefficient function vector is defined as

$$\mathbf{g}(k) = [\mathbf{g}_1(k)\ \ldots\ \mathbf{g}_{n_g}(k)] = [a_1\ \ldots\ a_{n_a}\ b_0\ \ldots\ b_{n_b}], \tag{7.5}$$

with $n_g = n_a + n_b + 1$, and

$$y(k) = \mathbf{g}(p(k))\mathbf{x}(k) + e(k). \tag{7.6}$$

Equation (7.6) can be rewritten as

$$y(k) = \sum_{i=1}^{n_g} \mathbf{g}_i(p(k)) x_i(k) + e(k), \tag{7.7}$$

where $x_i(k)$ indicates the i-th entry of the vector $\mathbf{x}(k)$ and $\mathbf{g}_i$ is the i-th entry of the vector $\mathbf{g}$.

7.3 INTRODUCTION TO GAUSSIAN PROCESSES

Gaussian process (GP) has been introduced to capture functional maps from observations and find the posterior distributions of the underlying functional dependencies over the observed data. The GP regression model in the dynamic case can be described as

$$y(k) = \mathcal{F}(\mathcal{D}(k)) + e(k), \tag{7.8}$$

where $\mathcal{D}(k)$ is the vector of the observations, and e is an i.i.d. noise process with $e(k) \sim \mathcal{N}(0, \sigma_e^2)$, denoting a normal distribution with zero mean and variance σ_e^2. In the LPV context, the GP method is adopted to estimate the model coefficients $\mathbf{g}$ and their dependencies on p assuming that $\mathcal{F}$ is describable as a particular realization of a Gaussian process with a zero-mean prior function and a previously chosen symmetric positive definite covariance function $\mathcal{K}(\cdot, \cdot)$ as

$$\mathcal{F}(\cdot) = \mathcal{GP}\big(0, \mathcal{K}(\cdot, \cdot)\big), \tag{7.9}$$

where $\mathcal{GP}$ denotes the Gaussian process [119]. Accordingly, the joint distribution of the output data y (conditioned) w.r.t. a given data set $\mathcal{D}$ and a test output data $\mathcal{F}_*$ is

$$\begin{bmatrix} y \\ \mathcal{F}_* \end{bmatrix} = \mathcal{N}\Bigg(0, \begin{bmatrix} \mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N & \mathcal{K}(\mathcal{D}, \mathcal{D}_*) \\ \mathcal{K}(\mathcal{D}_*, \mathcal{D}) & \mathcal{K}(\mathcal{D}_*, \mathcal{D}_*) \end{bmatrix}\Bigg), \tag{7.10}$$

where $\mathcal{D}_*$ is a given set of test points and $\mathcal{D}$ is the given set of observations. It should be noted that if there are N_* test points and N training data, then the covariance matrix $\mathcal{K}(\mathcal{D}, \mathcal{D}_*)$ would be an $N \times N_*$ matrix. Hence, to obtain the posterior distribution over functions, the joint distribution is conditioned on the observations. The following predictive equations can be obtained by deriving the conditional distributions [119]

$$\mathcal{F}_* | (\mathcal{D}, y, \mathcal{D}_*) \sim \mathcal{N}(\bar{\mathcal{F}}_*, \mathrm{Cov}(\mathcal{F}_*)), \tag{7.11}$$

$$\bar{\mathcal{F}}_* \triangleq \mathbb{E}[\mathcal{F}_* | (\mathcal{D}, y, \mathcal{D}_*)] = \mathcal{K}(\mathcal{D}_*, \mathcal{D})[\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N]^{-1} y, \tag{7.12}$$

$$\mathrm{Cov}(\mathcal{F}_*) = \mathcal{K}(\mathcal{D}_*, \mathcal{D}_*) - \mathcal{K}(\mathcal{D}_*, \mathcal{D})[\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N]^{-1} \mathcal{K}(\mathcal{D}, \mathcal{D}_*). \tag{7.13}$$

The mean and covariance functions obtained from (7.12)-(7.13) can statistically characterize the coefficients of the LPV model (7.7).

7.4 Formulation of the Error in Scheduling Variables

Gaussian processes have been successfully applied to a variety of applications in the context of dynamic systems and proven that they can accurately capture the underlying mapping of the input space to the output space. However, there are some limitations due to the assumptions made about the noise conditions. The standard GP algorithm is based on the assumption that the input data are noise free and the output data are corrupted by white (stationary) Gaussian noise. However, it is very likely – specially in industrial processes – that the input data are also corrupted by signal-independent sensor noise. As described earlier, in the present work, the scheduling variables are assumed to be corrupted with an i.i.d. Gaussian noise, in which the proposed identification approach can estimate the LPV coefficients in the presence of noise in the observations, i.e., noisy scheduling variables and outputs. Let p denote the n_p-dimensional scheduling variable vector defined as

$$\breve{p}_k = p_k + \varepsilon_p(k), \tag{7.14}$$

where $\varepsilon_p \sim \mathcal{N}(0, \Sigma_p)$ is white Gaussian noise that is independent of u and e. p is the noise-free scheduling variable that actually affects the underlying system. To simplify the notation, p_k is used instead of $p(k)$. It is assumed that the scheduling variables are independently corrupted by noise, and hence the noise variance Σ_p is a diagonal matrix. In the LPV model (7.7), the coefficients are functions of the noisy scheduling variables and calculating the posterior distribution is intractable using the standard GP framework. We employ the first order approximation of the model coefficients obtained using Taylor's series expansion on the observed data as

$$\mathbf{g}_i(\breve{p}_k - \varepsilon_p(k)) \approx \mathbf{g}_i(\breve{p}_k) - \varepsilon_p^\top \frac{\partial \mathbf{g}_i(\breve{p}_k)}{\partial p} + \cdots . \tag{7.15}$$

Since the derivative of a Gaussian process is itself another Gaussian process [133], the previous assumptions still hold true for the Taylor's expansion of the coefficients. However, one might argue that these functions are not available and need to be identified, which will be later examined in this section. Approximation in (7.15) gives a good estimation of the effect of scheduling variables noise on the function evaluation. We note that additional terms can also be kept beyond the affine

approximation in (7.15) at the expense of more complexity eventually leading to a much higher computational load. Substituting (7.15) into (7.7) and considering the derivative of the expectation (mean) of the LPV model coefficients obtained by Gaussian process, we have

$$y(k) = \sum_{i=1}^{n_g} \mathbf{g}_i(\breve{p}_k)x_i(k) - \sum_{i=1}^{n_g} \varepsilon_p^\top \frac{\partial \bar{\mathbf{g}}_i(\breve{p}_k)}{\partial p} x_i(k) + e(k), \tag{7.16}$$

where $\bar{\mathbf{g}}_i(\breve{p}_k)$ represents the mean value of the LPV model coefficients at an observed scheduling variable. The obtained heteroscedastic model considers the errors in both noisy output and noisy scheduling variables. Hence, the new error term can be considered as

$$\tilde{e}(k) = -\sum_{i=1}^{n_g} \varepsilon_p^\top \frac{\partial \bar{\mathbf{g}}_i(\breve{p}_k)}{\partial p} x_i(k) + e(k) \tag{7.17}$$

According to (7.16) and (7.17), the probability of the output y given the functions $\mathbf{g}_i, i = 1, \ldots, n_g$, and data set $\mathcal{D} = \{\breve{p}(k), y(k), u(k)\}_{k=1}^N$ can be obtained as

$$P(y|\ (\mathbf{g}, \mathcal{D})) = \mathcal{N}\big(\mathbb{E}(y), \sum_{i=1}^{n_g} x_i(m) \frac{\partial \bar{\mathbf{g}}_i(\breve{p}_m)}{\partial p}^\top \Sigma_p \frac{\partial \bar{\mathbf{g}}_i(\breve{p}_n)}{\partial p} x_i(n) + {\sigma_e}^2\big), \tag{7.18}$$

$$\mathbb{E}(y) = \sum_{i=1}^{n_g} \mathbf{g}_i(\breve{p}_k)x_i(k). \tag{7.19}$$

This can be seen as an equivalent formulation to considering the given scheduling variables as deterministic and adding a corrective term to the output error term. To obtain the posterior distribution, the prior is considered as the standard GP (7.8)

$$P\big(\mathbf{g}_i(p_k)x_i(k)|\mathcal{D}\big) = \mathcal{N}\big(0, \mathcal{K}^i(\mathcal{D}, \mathcal{D})\big), \tag{7.20}$$

where $\mathcal{K}^i(\mathcal{D}, \mathcal{D})$ is the $N \times N$ symmetric covariance matrix defined as

$$\mathcal{K}(\mathcal{D}_m, \mathcal{D}_n) = \sum_{i=1}^{n_g} x_i(m) k^i(\breve{p}_m, \breve{p}_n) x_i(n), \tag{7.21}$$

$$k^i(\breve{p}_m, \breve{p}_n) = \lambda_i \exp\left((\breve{p}_m - \breve{p}_n)^\top \mathcal{W}_i^{-1} (\breve{p}_m - \breve{p}_n)\right), \tag{7.22}$$

where $\mathcal{W}_i$ is the diagonal matrix of characteristic length-scale, and λ_i is a positive scalar factor representing the value of the covariance function when $\breve{p}_m$ and $\breve{p}_n$ are very close. Using the (approximation) LPV model in (7.16), the gradient term can be considered as a secondary error term to

compensate for the error in the scheduling variables and its effect on the output. Hence, similar to the variance of the output error, only the elements on the diagonal are kept for calculating the joint covariance matrix. The associated diagonal matrix is defined as

$$\mathscr{D}_i(n,n) = x_i(n)\frac{\partial \bar{\mathbf{g}}_i(\breve{p}_n)}{\partial p}^\top \Sigma_p \frac{\partial \bar{\mathbf{g}}_i(\breve{p}_n)}{\partial p} x_i(n), \tag{7.23}$$

for $n = 1, \ldots, N$. The calculated probabilities (7.18) and (7.20) are combined to obtain the following posterior distribution

$$\begin{bmatrix} y \\ \mathbf{g}_i(\mathcal{P}^*) \end{bmatrix} = \mathcal{N}\Bigg(0, \begin{bmatrix} \mathcal{K}(\mathcal{D},\mathcal{D}) + \sigma_e^2 I_N + \mathscr{D} & \kappa^i(\mathcal{D},\mathcal{P}^*) \\ \kappa^i(\mathcal{P}^*,\mathcal{D})^\top & k^i(\mathcal{P}^*,\mathcal{P}^*) \end{bmatrix}\Bigg), \tag{7.24}$$

where $\mathcal{P}^* = \breve{p}_i^*,\ i = 1, \ldots, N_*$, is a test point and $\kappa^i(\mathcal{D},\mathcal{P}^*)$ is defined as follows

$$\kappa^i(\mathcal{P}^*,\mathcal{D}) = \kappa^i(\mathcal{D},\mathcal{P}^*) = \begin{bmatrix} x_i(1)k^i(\breve{p}_1,\mathcal{P}^*) \\ x_i(2)k^i(\breve{p}_2,\mathcal{P}^*) \\ \vdots \\ x_i(N)k^i(\breve{p}_N,\mathcal{P}^*) \end{bmatrix}, \tag{7.25}$$

and $\mathscr{D}$ is the $N \times N$ matrix of the derivatives calculated as

$$\mathscr{D} = \sum_{i=1}^{n_g} \mathscr{D}_i, \tag{7.26}$$

and $\mathrm{Cov}(\mathbf{g}_i(\breve{p}_m), \mathbf{g}_i(\breve{p}_n)) = k^i(\breve{p}_m, \breve{p}_n)$ is the covariance or *kernel function* given by (7.22). According to (7.24), the predictive posterior mean and covariance are obtained as

$$\bar{\mathbf{g}}_i = \mathbb{E}[\mathbf{g}_i(\mathcal{P}^*)|\mathcal{D},\mathcal{P}^*] = \kappa^i(\mathcal{P}^*,\mathcal{D})\big[\mathcal{K}(\mathcal{D},\mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\big]^{-1}\mathcal{Y}, \tag{7.27}$$

$$\mathrm{Cov}[\mathbf{g}_i(\mathcal{P}^*)] = k^i(\mathcal{P}^*,\mathcal{P}^*) - \kappa^i(\mathcal{P}^*,\mathcal{D})\big[\mathcal{K}(\mathcal{D},\mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\big]^{-1}\kappa^i(\mathcal{D},\mathcal{P}^*), \tag{7.28}$$

where $\mathcal{Y} = [y(1), y(2), \ldots, y(N)]^\top$. To simplify the notation we define α as

$$\alpha = \Big(\mathcal{K}(\mathcal{D},\mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\Big)^{-1}\mathcal{Y}. \tag{7.29}$$

As observed, $\bar{\mathbf{g}}_i$ is dependent on its derivative, and hence an analytical solution does not exist to the resulting equations. Hence, an iterative procedure is proposed here. To this purpose, we first

calculate α using standard GP, i.e., from (7.29) without the derivative term $\mathscr{D}$. Then, the $\mathscr{D}_i$'s are obtained by substituting α into (7.30). $\mathscr{D}$ would then be computed from (7.26) and replaced in (7.29) to find α. This procedure is repeated until it converges to the best estimation of the system output through estimating the coefficient functions of the LPV model. The measure of the fitness along with more technical details are explained later in this section. After calculating the derivatives, substituting them in (7.23) and defining $\mathcal{P} = \{\breve{p}_1, \breve{p}_2, \ldots, \breve{p}_N\}$, we obtain

$$\mathscr{D}_i(n,n) = x_i(n)\alpha^\top \big(2\Delta_n \odot \kappa^i(\breve{p}_n, \mathcal{P})^\top\big)^\top \mathcal{W}_i^{-\top} \Sigma_p \mathcal{W}_i^{-1} \big(2\Delta_n \odot \kappa^i(\breve{p}_n, \mathcal{P})^\top\big)\alpha x_i(n), \tag{7.30}$$

where Δ_n is defined as $\Delta_n = [\breve{p}_n - \breve{p}_1, \breve{p}_n - \breve{p}_2, \ldots, \breve{p}_n - \breve{p}_N]^\top$ and $\kappa^i(\breve{p}_n, \mathcal{P})$ is a vector defined as

$$\kappa^i(\breve{p}_n, \mathcal{P}) = \Big[x_i(1)k^i(\breve{p}_n, \breve{p}_1)) \, , \cdots \, , \, x_i(N)k^i(\breve{p}_n, \breve{p}_N)\Big]. \tag{7.31}$$

It should be noted that for calculating the covariance matrix, the coefficient functions are assumed to be mutually independent and hence their associated derivatives are also mutually independent [133]. The added diagonal matrix $\mathscr{D}$ to the output noise variance in (7.24) is a corrective term that compensates for the error in the scheduling variables by taking into account the effect of the gradient of the mean function as a measure of sensitivity to noise-corrupted scheduling variables. Since, the corrective term $\mathscr{D}$ needs to be found to calculate the expectation of the LPV model coefficients, an iterative procedure is defined. First, the gradient of the estimated coefficients $\bar{\mathbf{g}}_i(p_k)$ by the standard GP are calculated and substituted in (7.24). In fact, we find the derivative of the coefficients from the mean function obtained via the standard GP at the training points. The obtained gradient is used to calculate the corrective additive term to update the probability distribution (7.24). Next, the updated distribution is used to estimate the coefficient functions and system output accordingly. Then, the hyperparameters including the noise variance of n_p scheduling variables and that of the output are tuned through trial and error to maximize the so-called *best fit ratio* (BFR) defined by

$$\text{BFR} := 100\% \cdot \max\left(1 - \frac{\|y(k) - \hat{y}(k)\|_{l_2}}{\|y(k) - \bar{y}\|_{l_2}}, 0\right), \tag{7.32}$$

which is considered to be the fitness score. In (7.32), $\hat{y}$ is the simulated output of the estimated model, y is the true output and $\bar{y}$ represents the mean of the true output y. Next, the gradient of the

estimated posterior mean $\bar{\mathbf{g}}_i(p_k)$ is used to update the corrective term and retrain the process. The procedure is iterative and continues until the maximum BFR is achieved.

7.4.1 Learning with Uncertain Scheduling Variables

The expectation (7.27) and covariance (7.28) of the coefficient functions obtained by GP are modified to include the scheduling variables noise contribution. To this aim, the expectation of the modified mean and covariance are obtained by integrating over the scheduling variables distribution [47]. In the present work, the test points are assumed to be a set of Gaussian distributions, and hence, the integral is analytically tractable. It should be noted that the true scheduling variables are not observable; however, we have access to their distribution $\mathcal{N}(\mathcal{P}^*, \Sigma_p)$, where $\mathcal{P}^*$ is the observed test point [33, 117]. Therefore, the noise-free scheduling variables are assumed to be Gaussian distributed $\tilde{\mathcal{P}}^* \sim \mathcal{N}(\mathcal{P}^*, \Sigma_p)$, where $\tilde{\mathcal{P}}^* = p_i^*,\ i = 1, \ldots, N_*$. According to the given distribution on the scheduling variables, the expectation of the covariance function (7.22) is obtained as

$$k_*^i(\mathcal{P}^*, \breve{p}_k) = \mathbb{E}_{\tilde{\mathcal{P}}^*}[k^i(\tilde{\mathcal{P}}^*, \breve{p}_k)|\mathcal{P}^*, \Sigma_p] = \int_{-\infty}^{+\infty} k^i(\tilde{\mathcal{P}}^*, \breve{p}_k) P(\tilde{\mathcal{P}}^*|\mathcal{P}^*, \Sigma_p) \mathrm{d}\tilde{\mathcal{P}}^*. \tag{7.33}$$

Eventually, we have

$$k_*^i(\mathcal{P}^*, \breve{p}_k) = \lambda_i \mid I + W_i^{-1}\Sigma_p \mid^{-\frac{1}{2}} \exp\bigg(- (\mathcal{P}^* - \breve{p}_k)^\top (\mathcal{W}_i + \Sigma_p)^{-1} (\mathcal{P}^* - \breve{p}_k)\bigg). \tag{7.34}$$

The expected value of the LPV model coefficients given the observed scheduling variables is obtained from (7.24) and (7.33) as

$$\mathbb{E}_{\tilde{\mathcal{P}}^*}[\mathbf{g}_i(\tilde{\mathcal{P}}^*) \mid (\mathcal{P}^*, \mathcal{D})] = \bar{\mathbf{g}}_i(\mathcal{P}^*) = \kappa_*^i(\mathcal{P}^*, \mathcal{D})^\top \bigg(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D} \bigg)^{-1} \mathcal{Y}, \tag{7.35}$$

where $\mathcal{Y} = [y(1)\ y(2)\ \ldots\ y(N)]$ and $\kappa_*^i(\mathcal{P}^*, \mathcal{D})$ is as defined in (7.25) considering the expectation of the covariance function $k_*^i(\mathcal{P}^*, \breve{p}_k)$ instead of the previously defined $k^i(\mathcal{P}^*, \breve{p}_k)$. To calculate the predictive covariance, the total covariance law is implemented (see [31]), where

$$\mathrm{Cov}[\mathbf{g}_i(\tilde{\mathcal{P}}^*) \mid (\mathcal{P}^*, \mathcal{D})] = \mathbb{E}_{\tilde{\mathcal{P}}^*}[\mathrm{Cov}[\mathbf{g}_i(\tilde{\mathcal{P}}^*) \mid (\mathcal{P}^*, \mathcal{D})]] + \mathrm{Cov}_{\tilde{\mathcal{P}}^*}[\bar{\mathbf{g}}_i(\tilde{\mathcal{P}}^*)]. \tag{7.36}$$

The covariance and mean are calculated from the distribution described by (7.24). Then, the total covariance law results in the following

$$\text{Cov}[\mathbf{g}_i(\tilde{\mathcal{P}}^*) \mid (\mathcal{P}^*, \mathcal{D})] = \mathbb{E}_{\tilde{\mathcal{P}}^*}[k^i(\tilde{\mathcal{P}}^*, \tilde{\mathcal{P}}^*)] - \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top \big(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I + \mathscr{D}\big)^{-1} \kappa^i(\mathcal{D}, \tilde{\mathcal{P}}^*)] + \mathbb{E}_{\tilde{\mathcal{P}}^*}[\bar{\mathbf{g}}_i(\tilde{\mathcal{P}}^*)\bar{\mathbf{g}}_i(\tilde{\mathcal{P}}^*)^\top] - \mathbb{E}_{\tilde{\mathcal{P}}^*}[\bar{\mathbf{g}}_i(\tilde{\mathcal{P}}^*)]\mathbb{E}_{\tilde{\mathcal{P}}^*}[\bar{\mathbf{g}}_i(\tilde{\mathcal{P}}^*)]^\top. \quad (7.37)$$

After substituting the predictive mean in (7.37), the predictive covariance function for a given test point $\mathcal{P}^*$ is obtained as

$$\begin{aligned}\text{Cov}[\mathbf{g}_i(\tilde{\mathcal{P}}^*) \mid (\mathcal{P}^*, \mathcal{D})] &= \mathbb{E}_{\tilde{\mathcal{P}}^*}[k^i(\tilde{\mathcal{P}}^*, \tilde{\mathcal{P}}^*)] - \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top \big(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\big)^{-1} \kappa^i(\mathcal{D}, \tilde{\mathcal{P}}^*)] \\ &+ \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top \bigg(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\bigg)^{-1} \mathcal{Y}\mathcal{Y}^\top \bigg(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\bigg)^{-1} \kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})] \\ -\mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top &\bigg(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\bigg)^{-1} \mathcal{Y}] \times \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top \bigg(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\bigg)^{-1} \mathcal{Y}].\end{aligned} \quad (7.38)$$

This can be rewritten in the following form

$$\text{Cov}[\mathbf{g}_i(\mathcal{P}^*)] = \lambda_i - \sum_{n=1}^{N}\sum_{m=1}^{N} \mathcal{S}_{mn} \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D}_m)\kappa^i(\mathcal{D}_n, \tilde{\mathcal{P}}^*)] + \mathcal{Y}^\top \mathcal{S} \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top]\mathcal{S}\mathcal{Y} - \bar{\mathbf{g}}_i(\mathcal{P}^*)^2, \quad (7.39)$$

where

$$\mathcal{S} = \big(\mathcal{K}(\mathcal{D}, \mathcal{D}) + \sigma_e^2 I_N + \mathscr{D}\big)^{-1}, \quad (7.40)$$

and

$$\mathbb{E}_{\tilde{\mathcal{P}}_*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top] = \int_{-\infty}^{+\infty} \kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top P(\tilde{\mathcal{P}}^* | \mathcal{P}^*, \Sigma_p) \mathrm{d}\tilde{\mathcal{P}}^*. \quad (7.41)$$

The integration over the given distribution leads to the following expression for the corresponding elements of (7.41)

$$\begin{aligned}\mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D})^\top]_{m,n} &= \mathbb{E}_{\tilde{\mathcal{P}}^*}[\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D}_m)\kappa^i(\tilde{\mathcal{P}}^*, \mathcal{D}_n)] = \lambda_i \mid 2\mathcal{W}_i^{-1}\Sigma_p + I \mid^{-\frac{1}{2}} \\ x_i(m)k^i(\mathcal{P}^*, \mathcal{D}_m)x_i(n)k^i(\mathcal{P}^*, \mathcal{D}_n) &\times \exp\bigg(- (\mathcal{P}^* - \frac{\breve{p}_m + \breve{p}_n}{2})^\top (\mathcal{W}_i + \frac{1}{2}\mathcal{W}_i\Sigma_p^{-1}\mathcal{W}_i)^{-1} \\ &(\mathcal{P}^* - \frac{\breve{p}_m + \breve{p}_n}{2})\bigg).\end{aligned} \quad (7.42)$$

Substituting (7.42) back into (7.39), the predictive covariance of the LPV model coefficients is obtained. Therefore, (7.35) together with (7.39) form the basis for *one-step ahead prediction* of the system output by the obtained predictive distribution over the given uncertain scheduling variables.

7.5 Two-step Procedure for LPV Model Identification with Scheduling Variables Denoising

A two-step training procedure is proposed here such that the kernels are first trained by the given validation data. The outcomes of the first step are the estimated hyperparameters including the noise variances σ_e^2 and Σ_p and the associated kernel parameters. As described later, the variance Σ_p is used to denoise the scheduling variables as the second step in the training process. In fact, the same estimated hyperparameters along with the adjusted kernel parameters are used to estimate the noise-free scheduling variables in time domain. The second step provides an accurate estimation of the scheduling variables that are implemented for approximating the system output and coefficient functions of the LPV model for the future test data.

Denoising the Scheduling Variables

The LPV scheduling variables are considered to be a noise corrupted time sequence signal. Considering (7.14), covariance of the noise corrupted scheduling variables is calculated as

$$\text{Cov}(\breve{p}(m), \breve{p}(n)) = \text{Cov}(p(m), p(n)) + \Sigma_p, \tag{7.43}$$

where $\breve{p}(m)$ and $p(m)$ denote the noisy and noise-free scheduling variables at given snapshot m, respectively. Also, Σ_p is the variance of the scheduling variables obtained in the previous section as one of the trained hyperparameters. We define a covariance kernel in time domain as

$$\text{Cov}(p_i(m), p_i(n)) = C(t_m, t_n) = \exp\left(\frac{\|t_m - t_n\|^2}{\sigma_i^2}\right), \tag{7.44}$$

where $i = 1, 2, \ldots, n_p$, $m, n \in \{1, 2, \ldots, N\}$ and t_m, t_n are m-th and n-th time snapshots, respectively. The posterior distribution of the scheduling variables is obtained as

$$p_i^* \triangleq \mathbb{E}[p_i^* | (t, \breve{p}_i, t^*)] = C(t^*, t)[C(t, t) + \sigma_{p_i}^2 I_N]^{-1} p_i, \tag{7.45}$$

where t^* is the time snapshot associated with the test point p_i^*. The kernel in (7.44) is tuned with by the same training data used for estimating the hyperparameters. The estimation of the scheduling variables from (7.45) is then utilized to estimate the model coefficients at the given test points. The estimated coefficients are used to estimate the output considering the same fitness score as given in (7.32) to realize the accuracy of the obtained estimation. It should be noted that the parameter σ_i is tuned to maximise the BFR as the measure of fitness.

7.6 Simulation Results and Discussion

In this section, two numerical examples are provided to examine the capability of the proposed approach in coping with uncertainties in the scheduling variables. Performance of the proposed approach in this paper is compared to the standard GP in the presence of a high level of noise in both scheduling variables and output measurements.

7.6.1 Example 1

An LPV system described by a finite impulse response (FIR) model and a nonlinear dynamic dependency on the scheduling variables is considered here. The model is described as

$$y(k) = \sum_{i=0}^{2} b_i(p_{k-i})u(k-i) + e_0(k), \tag{7.46}$$

with

$$b_0(p_k) = -\exp(-p_k), \quad b_1(p_{k-1}) = 1 + p_{k-1},$$
$$b_2(p_{k-2}) = \tan^{-1}(p_{k-2}),$$

where e_0 is a zero mean stochastic noise process with a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, $\sigma = 0.05$. The scheduling variable is generated by $p_k = \sin(\frac{\pi}{30}k)$ and an additive noise ε_p with a Gaussian distribution $\mathcal{N}(0, \Sigma_p)$, $\Sigma_p = 0.1$ is simulated to corrupt the scheduling variable resulting in $\breve{p}_k = p_k + \varepsilon_p(k)$,. A data set $\mathcal{D} = \{\breve{p}_k, y(k), u(k)\}_{k=1}^{N}$ with $N = 400$ snapshots is collected from the

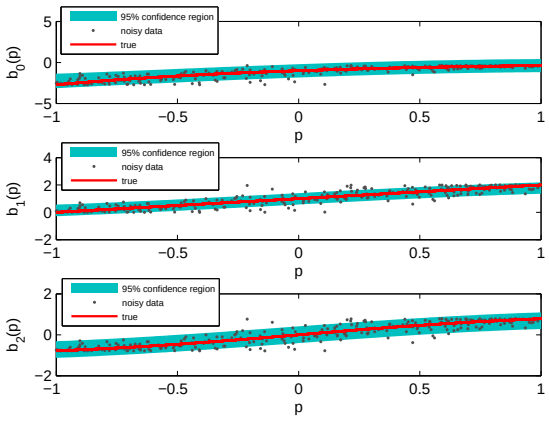


Figure 7.1: The estimated covariance functions by employing the proposed method.

system (7.46) by considering a periodic input $u(k)$ as

$$u(k) = \begin{cases} 1 & \text{if } k = 1 \\ 0 & \text{if } k = 2, 3. \end{cases} \tag{7.47}$$

As mentioned before, the training data set contains the noise corrupted scheduling variables and noisy output as the measurement data. The robustness of the proposed approach to the noise in variables is examined here and the results are compared to the standard GP. The hyperparameters including the RBF kernel parameters are obtained through the training process as $\mathcal{W}_1 = 1.17,\ \mathcal{W}_2 = 1.17,\ \mathcal{W}_3 = 1.17$ and $\lambda_1 = 1.2,\ \lambda_2 = 4.5,\ \lambda_3 = 5.6$; the output and scheduling variable's noise variance are also estimated as $\sigma_e = 0.048$ and $\Sigma_p = 0.98$, respectively. In addition, the time domain kernel in the denoising process is tuned with $\sigma_1 = 4.3$. The estimated covariance function for every coefficient using the proposed method and the standard GP are shown in figures 7.1 and 7.2, respectively.

As observed from the figures, the proposed approach provides a very good estimate of the level of uncertainty in the system with a much higher accuracy compared to the standard GP-based LPV model identification developed in [49]. The proposed approach offers promising results in estimating the uncertainty level by presenting a wider confidence region that contains the uncertain data, as well as the true coefficients, whereas the confidence regions obtained by the standard GP

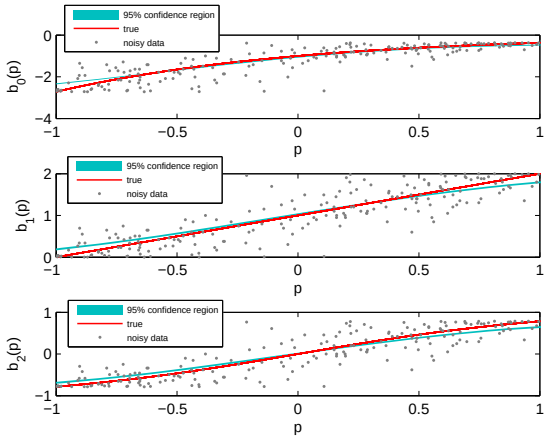


Figure 7.2: The estimated covariance functions using the standard GP as elaborated in [49].

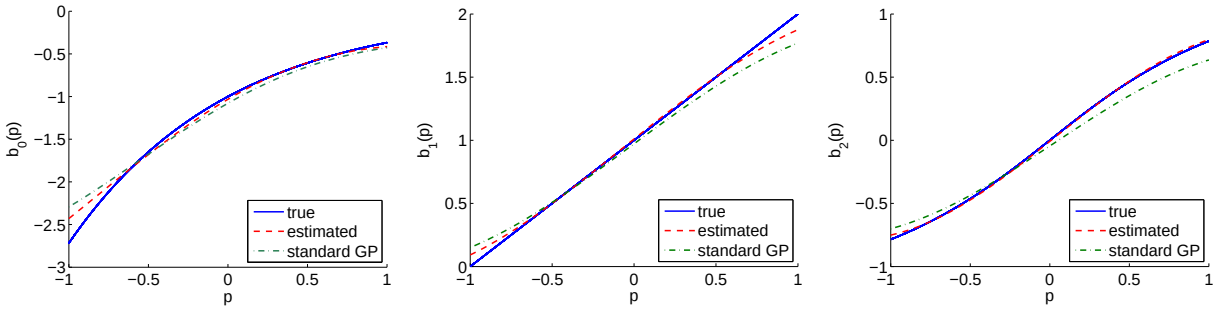


Figure 7.3: Estimated coefficient functions by the proposed method and standard GP.

do not include the true coefficients in various sections of the plots. This error can be justified due to the inability of standard GP in adapting to the presence of uncertainty in the scheduling variables. Furthermore, the estimated coefficient functions using the two approaches are shown in Figure 7.3. The *best fit ratio* (BFR) and *mean square error* (MSE) are used to quantify the estimated model accuracy and the results are shown in Table 7.1. The results illustrate that the proposed method in this paper can effectively provide an accurate estimation of the LPV model coefficients to cope with the uncertainty in the data.

Table 7.1: Example 1: The MSE and BFR of the estimated LPV model coefficient functions using the proposed LPV identification approach and the one in [49].

	Standard GP [49]		Proposed GP	
Coefficient	MSE	BFR	MSE	BFR
b_0	0.09	62.35%	0.0132	85.56%
b_1	0.0608	64.96%	0.0086	86.85%
b_2	0.0396	65.89%	0.0062	86.52%

7.6.2 EXAMPLE 2

An LPV data generating model in the ARX form is considered here to assess the performance of the proposed approach for rather more difficult nonlinearities to approximate. The system is described by

$$y(k) = -a_1(p(k))y(k-1) + \sum_{i=0}^{1} b_i(p_k)u(k-i) + e_0(k), \tag{7.48}$$

with

$$b_0(p(k)) = \begin{cases} +0.5 & \text{if } p(k) > 0.5 \\ p(k) & \text{if } -0.5 < p(k) < 0.5 \\ -0.5 & \text{if } p(k) < -0.5 \end{cases}$$

$$b_1(p(k)) = -0.2\,p^2(k),$$

$$a_1(p(k)) = \sin(p(k)),$$

where e_0 is a zero mean stochastic noise process with a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, $\sigma = 0.005$. The generated scheduling variable by $p_k = \sin(\frac{\pi}{101}k)$ is corrupted by an additive noise ε_p with a Gaussian distribution $\mathcal{N}(0, \Sigma_p)$, $\Sigma_p = 0.01$. A data set with $N = 600$ data points is generated by the system (7.48) with the periodic input $u(k)$ generated by (7.47). The hyperparameters are tuned by trial and error as $\mathcal{W}_1 = 2.7,\ \mathcal{W}_2 = 1.6,\ \mathcal{W}_3 = 2.8$ and $\lambda_1 = 1.2,\ \lambda_2 = 1.5,\ \lambda_3 = 1.6$. Moreover, the output and scheduling variable's noise variances are estimated as $\sigma_e = 0.0052$ and

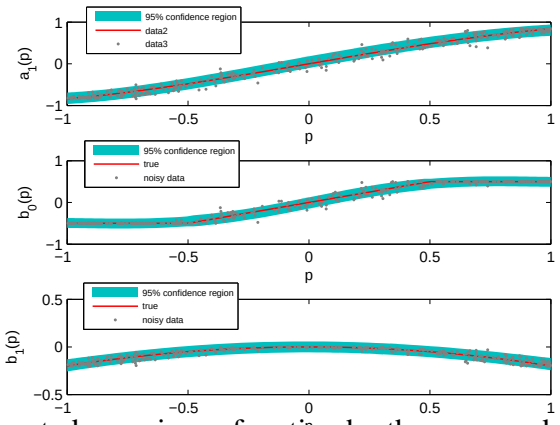


Figure 7.4: The estimated covariance function by the proposed method for Example 2.

$\Sigma_p = 0.011$. The estimated covariance functions that illustrate the associated confidence regions are shown in figures 7.4 and 7.5, for the presented GP and the standard GP in [49], respectively. The estimated coefficient functions are illustrated in Figure 7.6. Finally, Table 7.2 shows the BFR and MSE of the estimated LPV coefficients for the proposed LPV model identification approach and the one using the standard GP [49].

Table 7.2: Example 2: The MSE and BFR of the LPV model coefficients.

	Standard GP [49]		Proposed GP	
Coefficients	MSE	BFR	MSE	BFR
a_1	0.0081	83.02%	0.0009	94.78%
b_0	0.0032	84.97%	0.0011	92.15%
b_1	0.0064	73.72%	0.0016	90.9%

7.7 Concluding Remarks

A new system identification approach for input-output LPV models is presented in this paper based on Gaussian Process (GP) to compensate for the errors in the scheduling variables. The proposed approach uses a linear approximation to capture the effect of scheduling variables noise on the evaluated coefficient functions on the observed scheduling variables. This leads to acquiring a better

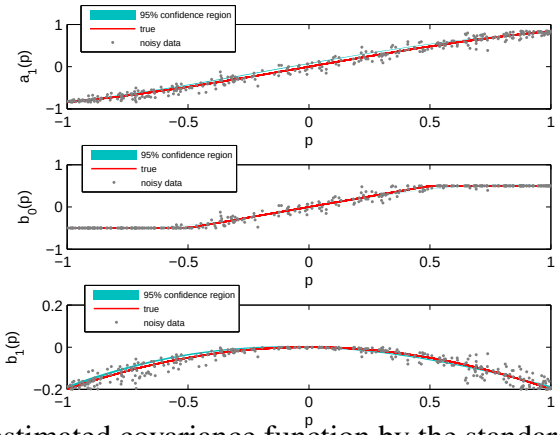


Figure 7.5: The estimated covariance function by the standard GP for Example 2.

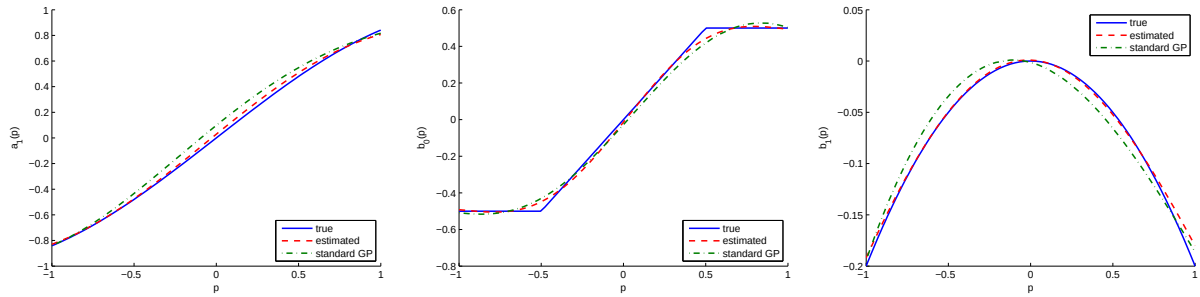


Figure 7.6: Coefficient functions in Example 2 estimated using the proposed method and the standard GP.

understanding of the uncertainties in data through more accurate formulation of the noise effect on the LPV model coefficients compared to the standard GP. The results indicate that the proposed method gives a more accurate estimation of the LPV model coefficient functions in the presence of both noisy measurement outputs and erroneous scheduling variables. The simulation results demonstrate that the proposed approach can effectively cope with uncertainties in the scheduling variables by estimating the variance of the noise due to the scheduling variables error and attempt to approximate the noise-free scheduling variables by implementing the time domain kernel functions.

CHAPTER 8

NONLINEAR MODEL ORDER REDUCTION OF BURGERS' EQUATION USING PROPER ORTHOGONAL DECOMPOSITION [1]

[1]F. Abbasi, J. Mohammadpour: Nonlinear Model Order Reduction of Burgers' Equation Using Proper Orthogonal Decomposition. 2015. In *Proc. of the American Control Conference*, Chicago, IL: pp. 583-588.

ABSTRACT

In this paper, we examine a model order reduction approach for dynamic systems governed by Burgers' equation with Neumann boundary conditions. The proper orthogonal decomposition (POD) method is employed here that provides a reliable and accurate modeling approach, while the temporal discretization of the continuous error function leads to a more accurate estimation of the defined cost function. We will investigate the accuracy of the reduced-order model compared to the finite element (FE) model by choosing an adequate number of basis functions for the approximating subspace. The derived lumped-parameter model for Burgers' equation is then described by a nonlinear state-space model. We finally demonstrate the accuracy of the reduced-order model through a numerical example, where we show that a 7-dimensional POD can accurately estimate the system output.

8.1 INTRODUCTION

Computational modeling and simulation of nonlinear complex, turbulent systems implementing the standard discretization schemes like finite element or finite difference, may require a large number of degrees of freedom to accurately describe the fluid flows. Consequently, the spatial discretization leads to scarce, but substantial nonlinear systems of ordinary differential equations (ODEs) that approximate the solution of the given system. However, with respect to both storage and computing time, these methods are inefficient. This can be crucial when the real-time solutions of complex systems in feedback control synthesis are required. As a remedy, the reduced-order modeling was introduced to describe the original mathematical model by a smaller model in a way that it can still represent certain significant aspects of the system or process with a good accuracy, depending on the order of the reduced model. That is to say, in implementing different model order reduction schemes, the lowest order of the reduced model, which accurately approximates the original system is desired. To achieve this, the original system or process should be described by a number of basis functions that are extracted from the expected solution of the system.

The proper orthogonal decomposition (POD), also known as the Karhunen-Loeve decomposition, can provide us with an effective tool based on projecting the dynamical system onto subspaces of basis elements that express characteristics of the given system. This is in contrast to, e.g., finite element techniques, where the elements are not correlated to the physical properties of the system they approximate [72, 105].

The implementation of the model order reduction approaches was originally developed by [94–96] in the framework of the structural simulation and later in simulation of incompressible viscous flows [149]. Among several commonly used model reduction techniques like balanced truncation and singular value decomposition based methods, the POD has received much attention in recent years as a tool to analyze complex physical systems [68, 122]. It was adopted by [129] to study turbulent flows. Another application of POD has been in the field of time-dependent partial differential equations (PDEs), where the snapshots are taken on a certain grid of time instants. It has been also successfully applied in different fields including signal analysis and pattern recognition [45], fluid dynamics and coherent structures [129], and more recently in optimal control of evolution problems [68].

The basis functions extracted by POD can be used in a collocation formulation of Galerkin projection that leads to a finite dimensional system with the smallest possible degrees of freedom. Therefore, the POD Galerkin technique is well suited in optimal control synthesis and the estimation of parameters in systems described by PDEs [67, 108]. The POD Galerkin scheme has been also extended separately for elliptic PDEs in [66]. Moreover, the application of POD Galerkin schemes for the spatial approximation has been substantially studied in [60, 64].

In the present work, we focus on the continuous POD (as opposed to snapshot POD) method and its application for the model order reduction of the forced Burgers' equation, which has characteristics similar to the Navier-Stokes equations. The objective of this work is to take advantage of the underlying characteristics of the continuous POD method to reduce the original model to a number of ODEs that would then be represented in the state-space form.

Throughout the paper, unless otherwise specified, we use the notation $\langle ., . \rangle$ to show the inner product of the given basis functions in the finite element method, representing the spatial domain integration of the product of the given basis functions. Also, $\mathcal{W}_i^l$ represents the i^{th} Fourier coefficient of the reduced model of the order l and $\mathbb{R}^m$ is an m-dimensional Euclidean space. In addition, we define $A \circ B$ as the Hadamard product of the matrices A and B, of the same dimension, $m \times n$ such that $[A \circ B]_{ij} = [A_{ij}][B_{ij}]$.

The rest of the paper is organized as follows: Section II describes the characteristics of the Burgers' PDE. Section III explains FEM and primary discretization of the system. The continuous POD and the fundamental idea behind it will be described in Section IV. Section V provides the obtained reduced-order models and their state-space representation. Finally, the simulation results for the given numerical example are discussed in section VI.

8.2 Nonlinear Parabolic Model of Burgers' Equation

Over the past three decades, Burgers' equation has been used for the better understanding of turbulence and other nonlinear phenomena as the very important parts of complex systems. This nonlinear parabolic partial deferential equation (PDE) provides a precise model for investigating different control problems such as boundary and distributed parameter feedback control problems. In the present study, we consider this nonlinear PDE model with Neumann boundary conditions aiming at developing a reduced-order and control-oriented model. In fact, we reduce this nonlinear PDE model to a number of ordinary differential equations (ODE), and then represent the system in state-space form using proper orthogonal decomposition (POD) method and finite element models (FEMs).

Suppose that Ω represents the spatial interval $(0, L)$ and that for $T > 0$, we set $Q = (0, T) \times \Omega$. For a given velocity $w(t, x)$ and viscosity ν, the governing viscous Burgers' PDE and the initial

and boundary conditions are

$$\frac{\partial w(t,x)}{\partial t} + w(t,x)\frac{\partial w(t,x)}{\partial x} - \nu\frac{\partial^2 w(t,x)}{\partial x^2} = f(t,x), \tag{8.1}$$

$$I.C: \quad w(0,x) = w_0(x), \tag{8.2}$$

$$B.C: \quad w_x(0,t) = u_1(t), w_x(L,t) = u_2(t) \tag{8.3}$$

where $(t,x) \in Q$, $u_1(t)$ and $u_2(t)$ are the changing boundary conditions or the system inputs, and the viscosity ν is considered as $1/Re$, where Re represents the Reynolds number. The function f is the forcing term that is assumed to be square integrable in space and time. We define the Hilbert space of Lebesgue square integrable functions as $H = L^2(\Omega)$. We note that the function f is in H if it satisfies

$$\int_0^T \|f(t,x)\|_H^2 \mathrm{d}t < \infty. \tag{8.4}$$

8.3 Order Reduction of Burgers' Equation with Finite Element Method

The finite element method (FEM) is considered as a general method to approximate partial deferential equations with lumped-parameter (ODE) models. An advantage of this technique over other methods is that if PDE is time dependent, then it can be reduced to a system of ODEs which can be integrated using existing techniques. Having a system of linear or nonlinear ODEs can allow to represent the system in the linear or nonlinear state-space form, which would be helpful for the control synthesis purposes.

FEM representation of the Burgers' equation

In order to discretize the Burgers' equation in the spatial domain, the interval is divided into N subintervals $[x_j, x_{j+1}]$ and we define $h_j = x_{j+1} - x_j$. We assume all the elements are of equal size (uniformly spaced mesh) and hence $h_1 = \ldots = h_N = h$. Therefore, the FEM basis functions are defined as [42]

(i) for elements $e_j,\ j = 1 : N-1$,

$$\mathcal{N}_j(x) = \begin{cases} \frac{x - x_{j-1}}{h},\ x_{j-1} \le x \le x_j \\ \frac{x_{j+1} - x}{h},\ x_j \le x \le x_{j+1} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{8.5}$$

(ii) for element e_0,

$$\mathcal{N}_0(x) = \begin{cases} \frac{x_1 - x}{h},\ 0 \le x \le x_1 \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{8.6}$$

(iii) for element e_N

$$\mathcal{N}_N(x) = \begin{cases} \frac{x - x_{N-1}}{h}, x_{N-1} \le x \le x_N \\ 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{8.7}$$

The approximation of $w(t,x)$ in the space spanned by the piecewise linear basis functions is given by

$$w^N(t,x) = \sum_{i=0}^{N} \mathcal{W}_i(t)\mathcal{N}_i(x), \tag{8.8}$$

where $\mathcal{W}_i(t)$ is the nodal value, i.e., $w(t,x_i)$ at the ith node and time *t* [42]. The *weak solution* approach is employed by multiplying both sides of (8.1) by a piecewise smooth test function $v(x)$ and integrating in the spatial variable's domain [19]. Taking the integral from both sides and also substituting the second order derivative term by the chain rule results in

$$\int_0^L \left(w_t(t,x) + \frac{1}{2}[w^2(t,x)]_x \right) v(x)\,\mathrm{d}x - \nu \left[u_2(t)v(L) - u_1(t)v(0) - \int_0^L w_x(t,x)v'(x)\,\mathrm{d}x \right] = \int_0^L f(t,x)v(x)\,\mathrm{d}x. \tag{8.9}$$

Using the group finite element (GFE) method proposed in [44], nonlinear term can be approximated as

$$w^2(t,x) \approx \sum_{i=0}^{N} \mathcal{W}_i^2(t)\mathcal{N}_i(x). \tag{8.10}$$

Since $v(x)$ is arbitrary and piecewise smooth, we let $v(x) = \mathcal{N}_j(x)$, for $j = 0, 1, \ldots, N$. Using Galerkin method and substituting (8.10) into (8.9) yields

$$\sum_{i=0}^{N} \dot{\mathcal{W}}_i(t) \int_0^L \mathcal{N}_i(x)\mathcal{N}_j(x)\,\mathrm{d}x + \frac{1}{2}\sum_{i=0}^{N} \mathcal{W}_i^2(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j(x)\,\mathrm{d}x - \nu\Big[u_2(t)\mathcal{N}_j(L) - u_1(t)\mathcal{N}_j(0)\Big] - \nu \sum_{i=0}^{N} \mathcal{W}_i(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j'(x)\,\mathrm{d}x = \int_0^L f(t,x)\mathcal{N}_j(x)\,\mathrm{d}x.$$

We consider the following notation

$$M_{ij} = \big\langle \mathcal{N}_i(x), \mathcal{N}_j(x) \big\rangle, S_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j'(x) \big\rangle,$$
$$F_j(t) = \big\langle f(t,x), \mathcal{N}_j(x) \big\rangle. \tag{8.11}$$

For the nonlinear term, we represent it as

$$(N(\mathcal{W}(t)))_j = \frac{1}{2}\sum_{i=0}^{N} \mathcal{W}_i^2(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j(x),$$
$$K_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j(x) \big\rangle,$$

Therefore

$$N(\mathcal{W}(t)) = \frac{1}{2} K \mathcal{W}^2(t), \tag{8.12}$$

where $\mathcal{W}^2(t) = [\mathcal{W}_0^2(t) \ \ldots \ \mathcal{W}_N^2(t)]^\top$. Finally, the reduced-order model with input vector $U(t) = [u_1(t) \ \ u_2(t)]^\top$ is described as a set of $N+1$ ordinary differential equations

$$M\dot{\mathcal{W}}(t) + \nu S\mathcal{W}(t) + \frac{1}{2}K\mathcal{W}^2(t) - \nu L U(t) = F(t), \tag{8.13}$$

where

$$L = \begin{bmatrix} -1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}^\top_{(N+1)\times 2}.$$

To solve this set of nonlinear ODEs, we need to specify the initial condition. To do so, the given initial condition should be described in the space spanned by the basis functions, i.e.,

$$w_0(x) \approx w^N(0,x) = \sum_{i=0}^{N} \mathcal{W}_i(0)\mathcal{N}_i(x). \tag{8.14}$$

By multiplying both sides of the equality by the test function and again using the weak solution approach, we obtain

$$\sum_{i=0}^{N} \mathcal{W}_i(0) \int_0^L \mathcal{N}_i(x)\mathcal{N}_j(x) = \int_0^L w_0(x)\mathcal{N}_j(x), \quad j = 0, \ldots, N. \tag{8.15}$$

This can be represented in the matrix form as

$$M\mathcal{W}(0) = \mathcal{P}, \tag{8.16}$$

where $\mathcal{P}_j = \langle w_0(x), \mathcal{N}_j(x) \rangle$. This linear equation gives the initial conditions for the $N + 1$ ODEs described by (8.13).

8.4 Proper Orthogonal Decomposition Method

The fundamental idea behind proper orthogonal decomposition (POD) is to optimally represent the system in a mean-squared error sense using an orthonormal basis of rank l. Let $Y = [y_1, \ldots, y_n]_{m\times n}$ be a real data matrix containing the n snapshot vectors of m spatial data points. The POD basis of rank l is optimal in the sense of representing the columns $\{y_j\}_{j=1}^n$ of Y as a linear combination by an orthonormal basis of rank l [72]. We endow the Euclidean space $\mathbb{R}^m$ with the weighted inner product as

$$\langle u, \tilde{u} \rangle_W = u^T W \tilde{u} = \langle u, W\tilde{u} \rangle_{\mathbb{R}^m} = \langle Wu, \tilde{u} \rangle_{\mathbb{R}^m} \quad \text{for } \; u, \tilde{u} \in \mathbb{R}^m \tag{8.17}$$

where $W \in \mathbb{R}^{m\times m}$ is a symmetric, positive-definite matrix. Note that the vector $y(t)$, $t\in [0, T]$, now represents a function in Ω evaluated at m grid points. Therefore, we should supply $\mathbb{R}^m$ by a weighted inner product representing a discretized inner product in an appropriate function space. Since, the mass matrix in (8.13) is symmetric, real and positive definite, it can be considered as the weight matrix in the predefined inner product. The goal is to determine a POD basis of rank $l \leq n$ that gives the best estimate of the entire trajectory $\nu^y = \text{span}\{y(t)|t \in [0, T]\} \subset \mathbb{R}^m$. The optimality is achieved by minimizing the error between the data and its projection onto the basis

set [64]

$$J = \min \int_0^T \|y(t) - \sum_{i=1}^{l} \langle y(t), \tilde{u}_i \rangle_M \tilde{u}_i\|_M^2 \, \mathrm{d}t \tag{8.18}$$
$$\text{s.t.} \quad \langle \tilde{u}_i, \tilde{u}_j \rangle_M = \delta_{ij} \quad \text{for } 1 \leq i, j \leq l.$$

Since the entire trajectory is not available in practical computation, we suppose that we know the solution of (8.13) at the given time instants t_j, $j = 1, \ldots, n$. This translates to minimizing the following cost function J while the constraints are met [66]

$$J = \min \sum_{j=1}^{n} \alpha_j \|y_j - \sum_{i=1}^{l} \langle y_j, \tilde{u}_i \rangle_M \tilde{u}_i\|_M^2 \tag{8.19}$$
$$\text{s.t.} \quad \langle \tilde{u}_i, \tilde{u}_j \rangle_M = \delta_{ij} \quad \text{for } 1 \leq i, j \leq l,$$

where α_j's denote the non-negative trapezoidal weights defined by

$$\alpha_1 = \frac{\Delta t}{2}, \ \alpha_j = \Delta t \quad \text{for } 2 \leq j \leq n-1, \ \alpha_n = \frac{\Delta t}{2}. \tag{8.20}$$

To solve the above constrained optimization problem, first-order necessary optimality condition is applied. Therefore, the associated Lagrange functional is described as

$$\mathcal{L} : \underbrace{\mathbb{R}^m \times \ldots \times \mathbb{R}^m}_{l-times} \times \mathbb{R}^{l \times l}$$
$$\mathcal{L}(u_1, \ldots, u_l, \Lambda) = \sum_{j=1}^{n} \alpha_j \|y_j - \sum_{i=1}^{l} \langle y_j, u_i \rangle_M u_i\|_M^2 + \sum_{i=1}^{l} \sum_{j=1}^{l} \Lambda_{ij} (1 - \langle u_i, u_j \rangle_M), \tag{8.21}$$

where $u_1, \ldots, u_l \in \mathbb{R}^m$ and $\Lambda \in \mathbb{R}^{l \times l}$. First-order necessary optimality condition gives

$$\Delta_{u_i} \mathcal{L}(u_1, \ldots, u_l, \Lambda) = 0 \ \text{ in } \mathbb{R}^m, \ 1 \leq i \leq l. \tag{8.22}$$

Also noting that

$$\langle u_i, u_j \rangle_M = \delta_{ij} \quad \text{for } 1 \leq i, j \leq l, \tag{8.23}$$

the following is derived from the optimality condition (8.22)

$$Y D Y^T M u_i = \lambda_i u_i \ \text{ for } i = 1, \ldots, l, \tag{8.24}$$

where $D = \text{diag}(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^{n\times n}$. By defining $u_i = M^{-1/2}\bar{u}_i$ in (8.24) and multiplying it by $M^{1/2}$ from left, we obtain

$$M^{1/2}YDY^TM^{1/2}\bar{u}_i = \lambda_i\bar{u}_i \;\text{ for } i = 1, \ldots, l. \tag{8.25}$$

Considering (8.23), we have

$$\langle \bar{u}_i, \bar{u}_j\rangle_{\mathbb{R}^m} = \bar{u}_i^T\bar{u}_j = u_i^TMu_j = \langle u_i, u_j\rangle_M = \delta_{ij} \quad \text{for } 1 \le i, j \le l. \tag{8.26}$$

Defining $\bar{Y} = M^{1/2}YD^{1/2} \in \mathbb{R}^{m\times n}$ and knowing that $M^T = M$ and $D^T = D$, the solution to the optimization problem (8.19) is obtained by solving the symmetric $m \times m$ eigenvalue problem

$$\begin{aligned} \bar{Y}\bar{Y}^T\bar{u}_i &= \lambda_i\bar{u}_i, \;\; 1 \le i \le l, \\ \langle \bar{u}_i, \bar{u}_j\rangle_{\mathbb{R}^m} &= \delta_{ij}, \;\; 1 \le i, j \le l. \end{aligned} \tag{8.27}$$

The choice of the number of basis functions l, leading to an accurate description of the original model, is certainly of critical importance for applying POD. There is no general rule for the selection of l; it is rather based on heuristic considerations along with observing the captured relative energy by the basis functions [21], which is expressed by

$$\mathcal{E}(l) = \frac{\sum_{i=1}^{l}\lambda_i}{\sum_{i=1}^{d}\lambda_i}. \tag{8.28}$$

where $d = rank(\bar{Y})$.

8.5 Development of the Reduced-order Model Using POD

In this section, the derivation of the reduced order model of the Burgers' equation using POD method is described. To obtain a control-oriented model, we use the approximation of $w(t, x)$ in the space spanned by the POD basis functions $\psi_i(x),\ i = 1, \ldots, l$ as

$$w(t, x) = \sum_{i=1}^{l}\langle w(t, x), \psi_i(x)\rangle_M\psi_i(x). \tag{8.29}$$

By setting

$$\mathcal{W}_i^l(t) = \langle w(t, x), \psi_i(x)\rangle_M, \tag{8.30}$$

we have the Galerkin form of the projection to the POD space, which is

$$w(t,x) = \sum_{i=1}^{l} \mathcal{W}_i^l(t)\psi_i(x), \tag{8.31}$$

where the Fourier coefficients $\mathcal{W}_i^l$, $1 \leq i \leq l$, are functions mapping $[0,T]$ onto $\mathbb{R}$. Since for $l = m$, we have $w(t,x) = w^l(t,x)$, it can be deduced that $w^l(t,x)$ gives an approximation of $w(t,x)$ provided that $l \leq m$.

We recall the weak solution approach described earlier for Burgers' equation that led to

$$\int_0^L \left(w_t(t,x) + \frac{1}{2}[w^2(t,x)]_x \right) v(x)\,\mathrm{d}x - \nu \left[u_2(t)v(L) - u_1(t)v(0) - \int_0^L w_x(t,x)v'(x)\,\mathrm{d}x \right] = \int_0^L f(t,x)v(x)\,\mathrm{d}x. \tag{8.32}$$

Since $v(x)$ is arbitrary and piecewise smooth, we let $v(x) = \psi_j(x), j = 1,2,...,l$ and use POD Galerkin projection that results in

$$\sum_{i=1}^{l} \dot{\mathcal{W}}_i^l(t) \int_0^L \psi_i(x)\psi_j(x)\,\mathrm{d}x + \frac{1}{2}\int_0^L \left(\left[\sum_{i=1}^{l} \mathcal{W}_i^l(t)\psi_i(x) \right]^2 \right)_x \psi_j(x)\,\mathrm{d}x - \nu \left[u_2(t)\psi_j(L) - u_1(t)\psi_j(0) \right] + \nu \sum_{i=1}^{l} \mathcal{W}_i^l(t) \int_0^L \psi_i'(x)\psi_j'(x)\,\mathrm{d}x = \int_0^L f(t,x)\psi_j(x)\,\mathrm{d}x.$$

Considering the following notation

$$M_{ij}^l = \langle \psi_i(x), \psi_j(x) \rangle,\ S_{ij}^l = \langle \psi_i'(x), \psi_j'(x) \rangle,$$
$$F_j^l(t) = \langle f(t,x), \psi_j(x) \rangle,$$
$$(N^l(\mathcal{W}^l(t)))_j = \frac{1}{2}\int_0^L N\big(w(t,x)\big)\psi_j(x)\,\mathrm{d}x,$$

reduced order model becomes

$$M^l\dot{\mathcal{W}}^l(t) + \nu S^l\mathcal{W}^l(t) + N^l(\mathcal{W}^l(t)) - \nu L^l U = F^l(t). \tag{8.33}$$

It is noted that when the basis functions are orthonormal, $M^l = I_r$. Matrix S^l in the reduced-order model can also be obtained from the original full-order matrices by expanding the POD basis

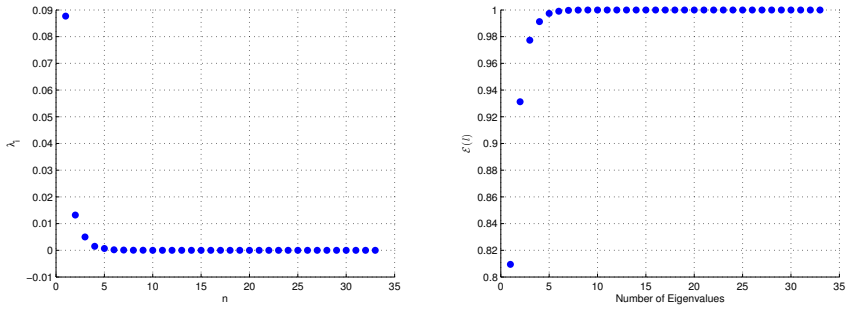


Figure 8.1: The extracted eigenvalues corresponding to the POD eigenvectors (left); The energy captured by different chosen number of eigenvalues (right).

functions as

$$\begin{aligned} S_{ij}^{l} &= \left\langle \psi_i'(x), \psi_j'(x) \right\rangle = \int_0^L \psi_i'(x)\psi_j'(x)\,\mathrm{d}x \\ &= \int_0^L \sum_{k=0}^{N} \Psi_{ki}\mathcal{N}_k'(x) \sum_{m=0}^{N} \Psi_{mj}\mathcal{N}_m'(x)\,\mathrm{d}x \\ &= \sum_{k=0}^{N}\sum_{m=0}^{N} \Psi_{ki}\Psi_{mj} \int_0^L \mathcal{N}_k'(x)\mathcal{N}_m'(x)\,\mathrm{d}x. \end{aligned} \tag{8.34}$$

This can be written in the following matrix form

$$S^{l} = (\Psi^{l})^{\top} S \Psi^{l}. \tag{8.35}$$

Hence, we have characterized matrix S^l in terms of full order matrix S and the POD basis functions. Same procedure can be implemented to represent the nonlinear term in the reduced-order model in terms of the full order matrices. To archive this, we first need to find the relationship between the coefficients $\mathcal{W}_1(t), \ldots, \mathcal{W}_N(t)$ and $\mathcal{W}_1^l(t), \ldots, \mathcal{W}_l^l(t)$. Considering the fact that the solution w can be expressed in either l-dimensional reduced-order space or $N+1$-dimensional full order system, we have

$$w(t,x) \approx \sum_{i=1}^{l} \mathcal{W}_i^l(t)\psi_i(x) \approx \sum_{i=0}^{N} \mathcal{W}_i(t)\mathcal{N}_i(x). \tag{8.36}$$

As described earlier, the POD basis functions can be written as a linear combination of the FE basis functions, and hence

$$\sum_{i=0}^{N} \mathcal{W}_i(t)\mathcal{N}_i(x) \approx \sum_{i=1}^{l} \mathcal{W}_i^l(t) \sum_{m=0}^{N} \Psi_{mi}\mathcal{N}'_m(x). \tag{8.37}$$

This describes the Fourier coefficients in the compact form as

$$\mathcal{W}(t) \approx \Psi^l \mathcal{W}^l(t). \tag{8.38}$$

Substituting (8.38) in (8.12) and using Hadamard product, we obtain

$$N^l(\mathcal{W}^l(t)) = \frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)). \tag{8.39}$$

We further define

$$F^l(t) = (\Psi^l)^\top F(t), \quad L^l = (\Psi^l)^\top L, \tag{8.40}$$

and hence the reduced order model is described by

$$\dot{\mathcal{W}}^l(t) + \nu S^l\mathcal{W}^l(t) + \frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)) - \nu L^l U = F^l(t). \tag{8.41}$$

STATE-SPACE REPRESENTATION

After deriving the reduced-order model for the Burgers' PDE in the form of $N+1$ ordinary differential equations in (8.13) using finite element method, we can describe the system of ODEs in the state-space form with $N+1$ states as

$$\dot{\mathcal{W}}(t) = A\mathcal{W}(t) + \mathbf{h}(t, \mathcal{W}(t), U(t)), \tag{8.42}$$

where

$$A = -\nu M^{-1}S, \ \mathbf{h}(t, \mathcal{W}(t), U(t)) = -\frac{1}{2}M^{-1}K\mathcal{W}^2(t) + M^{-1}F(t) + \nu M^{-1}LU(t).$$

The reduced order state-space model is then represented by

$$\dot{\mathcal{W}}^l(t) = A^l\mathcal{W}^l(t) + \mathbf{g}(t, \mathcal{W}^l(t), U(t)), \tag{8.43}$$

where

$$A^l = -\nu S^l, \ \mathbf{g}(t, \mathcal{W}^l(t), U(t)) = \ -\frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)) + F^l(t) + \nu L^l U(t).$$

8.6 Simulation Results and Discussions

In order to assess the performance of the presented model reduction method, an example of a viscous Burgers' equation is discussed in this section. Our goal is to determine whether the reduced-order model can accurately estimate the full-order FE model. In this example, the forcing term in (8.1) is considered to be zero that translates to the so-called viscous Burgers' equation. The initial condition is also assumed to be

$$w_0(x) = \begin{cases} 100(sin(8\pi x) - 2x), & \text{if } x \in (0, \frac{1}{4}] \\ 0, & \text{otherwise.} \end{cases} \tag{8.44}$$

It is also assumed that the boundary conditions, i.e., inputs to the state-space models, are sinusoidal functions as

$$u_1(t) = 0.8sin(3t), u_2(t) = 0.5sin(3t). \tag{8.45}$$

As the first step, the eigenvalues corresponding to the POD method are extracted. Figure 8.1 shows the eigenvalues in a descending order. Also, the percentage of the total energy captured by the chosen number of eigenvalues is shown in Figure 8.1. In order to investigate the accuracy of the implemented model order reduction approach, FEM and 7-dimensional POD solution of Burgers' equation is shown in Figure 8.2. According to Figure 8.1, the first 7 eigenvalues capture more than 99% of the system's total energy. This is observed in Figure 8.2, where the POD solution closely matches the FEM solution.

The reduced and full-order open-loop models are simulated for a given viscosity $\nu = 0.01$ (or *Re* = 100) to gauge the performance of the model reduction approach for a class of physical flows. Figures 8.3 illustrates the output of the FE and reduced order POD models for a sinusoidal input signal in the given example. It is observed that by increasing the number of basis functions to 7, we can achieve a very close match between the output of both models. Finally, to quantify the model accuracy, we consider the *Best Fit Rate* (BFR) defined as

$$\text{BFR} = \max\left(1 - \frac{\|y_i(k) - \hat{y}_i(k)\|_{l_2}}{\|y_i(k) - \bar{y_i}\|_{l_2}}, 0\right), \tag{8.46}$$

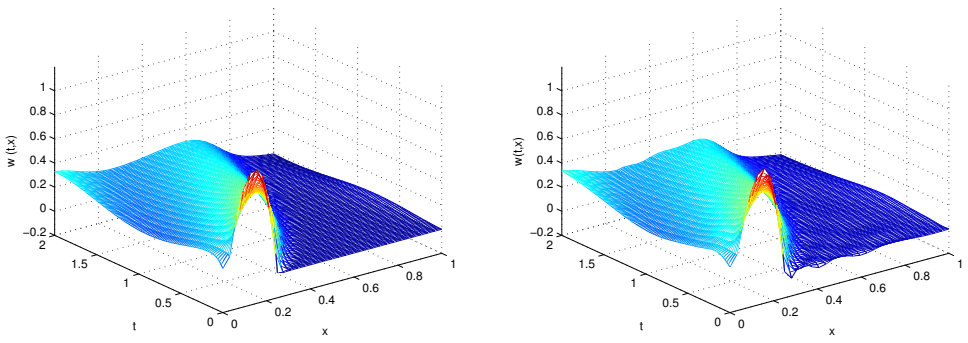


Figure 8.2: Finite element solution for 32 spatial points (left), and POD solution with 7 basis functions (right) both with $\nu = 0.01$.

where y_i and $\hat{y}_i$ represent the i^{th} output of the FEM and POD state-space models, respectively, and $\bar{y}_i$ is the mean value of the i^{th} output of the FEM model. In fact, the obtained BFR measures the matching between the output of the reduced order models and the output of the FE model. Table 8.1 shows the BFR of the reduced order models. Also shown in Table 8.1 is the mean-squared error (MSE) between the reduced-order model output and the original FE model output.

Table 8.1: The MSE and BFR of the output signal of the reduced order models with sinusoidal inputs

	POD (3 Bases)		POD (5 Bases)		POD (7 Bases)	
Output	MSE	BFR	MSE	BFR	MSE	BFR
y_1	0.0021	0.0886	2.1836e-04	0.7042	1.3275e-05	0.9271
y_2	1.0203e-04	0.4909	6.7584e-06	0.8690	4.2925e-06	0.8956

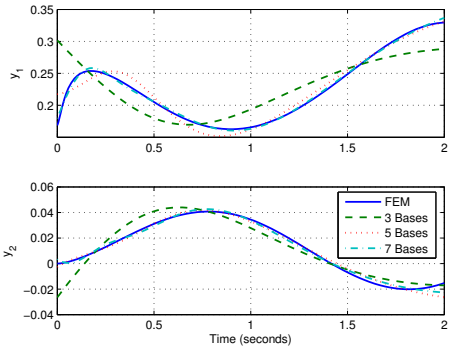


Figure 8.3: The output of the FE model and the reduced-order model obtained using POD for different number of basis functions.

8.7 Concluding Remarks

In this paper, we presented results on developing a reduced-order state-space model for a nonlinear parabolic PDE to overcome several drawbacks arising from implementing the full-order model specially for real-time applications. The model obtained used POD Galerkin method, and guaranteed a highly accurate approximation of the original model. We showed that the combination of POD and the weak solution approach can lead to an accurate reduced-order model. The results demonstrated that increasing the number of chosen eigenvalues would represent the original system with a higher accuracy. We are currently examining the design of robust nonlinear controllers based on the reduced-order nonlinear model derived in this paper.

CHAPTER 9

REDUCED ORDER MODEL-BASED SLIDING MODE CONTROL OF DYNAMIC SYSTEMS GOVERNED BY BURGERS' EQUATION [1]

[1] F. Abbasi, J. Mohammadpour: Reduced Order Model-based Sliding Mode Control of Dynamic Systems Governed by Burgers' Equation. 2015. *Proc. of the 54th IEEE Conference on Decision and Control*, Osaka, Japan: pp. 7380-7385.

ABSTRACT

In this paper, we use the reduced-order nonlinear model of dynamic systems governed by Burgers' equation with Neumann boundary conditions – recently developed by the authors in [4] – to define low order sliding mode surfaces. While keeping the system states moving on the defined surface, the imposed control law guarantees the stability of the full-order model obtained using a finite element (FE) approximation of the Burgers' equation. The accuracy of the applied reduced-order model obtained from proper orthogonal decomposition (POD) method compared to the FE model is investigated by determining an adequate number of basis functions for the approximating subspace. The reduced-order model is then used to design a sliding mode controller, which is implemented on the FE model demonstrating that the obtained reduced model is suitable for both stabilization of the full-order model and trajectory tracking.

9.1 INTRODUCTION

The implementation of the standard discretization methods such as finite element or finite difference methods may require a large number of degrees of freedom to accurately describe complex nonlinear partial differential equations (PDEs) like Burgers' equation. Accordingly, the control design task based on those discretized full-order models would be cumbersome. This can also be crucial when the real-time solutions for feedback control of complex systems are sought. As a remedy, the reduced-order modeling was introduced to approximate the original dynamic model by a simpler one so it could still represent certain significant aspects and dominant dynamics of the system or process with an acceptable accuracy depending on the complexity of the reduced-order model. That is to say in implementing different model order reduction schemes, a model with the lowest order, which accurately approximates the original full-order model is desired. To achieve this, the original process should be initially described by a number of basis functions extracted from the expected solution of the system. The development of the model order reduc-

tion approaches was proposed in [94] in the framework of the structural simulation and later for simulation of incompressible viscous flows [149].

Recent emergence of data analysis techniques have seen an increased use of tools like principal component analysis (PCA) [120] and proper orthogonal decomposition (POD) for developing control-oriented reduced models of nonlinear systems; a recent example of this is our earlier work [4] that developed a reduced-order model of Burgers' equation using continuous POD. The developed reduced-order model forms the basis for the design of a robust controller in this paper, that is capable of handling possible model uncertainties. Hence, a nonlinear control strategy based on the reduced-order sliding mode is proposed here to tackle different kinds of uncertainties arising from parametric and modeling imprecisions in the reduced-order nonlinear model of Burgers' equation. Sliding mode control (SMC) is a nonlinear feedback control scheme that can effectively apply a high-frequency switching control to alter the dynamics of a nonlinear system [112]. Switching from one continuous mode to another considering the system's current position in state space can guarantee the convergence of the trajectories towards a switching surface that eventually slides along the boundaries of the control structures [36]. For the design of a sliding mode controller, Lyapunov stability theory is utilized to guarantee the stability of the full-order nonlinear model by defining the reduced-order sliding surfaces. The model order reduction methods can be crucial in coping with high order models, in which the measurements of the system states are required. This translates to high computational complexity in both controller and observer design. More specifically, the high dimensional matrix operations including inversion involved in higher order models can considerably increase the run time [17].

In the present work, we employ the POD-based reduced-order model of the Burger's equation developed recently by the authors in [4] for the design of a sliding mode controller that can guarantee the stability of the full-order finite element (FE) model. The closed-loop performance achieved by the reduced-order model based sliding mode controller in the presence of modeling uncertainties demonstrates that the designed nonlinear controller can effectively manipulate the

full-order model. This can significantly decrease the computational load to control the high order model.

Throughout the paper, unless otherwise specified, notation $\langle ., . \rangle$ represents the inner product of the given basis functions in the finite element method, which is the spatial domain integration of the product of the given basis functions. Also, $\mathcal{W}_i^l$ represents the i^{th} Fourier coefficient of the reduced-order model of order l and $\mathbb{R}^m$ is the m-dimensional Euclidean space. Moreover, Kronecker delta, δ_{ij}, returns zero for $i \neq j$ and 1 for $i = j$. Finally, we define $A \circ B$ as the Hadamard product of the matrices A and B of the same dimension such that $[A \circ B]_{ij} = [A_{ij}][B_{ij}]$.

The rest of the paper is organized as follows. Section 9.2 describes the Burgers' PDE and its finite element modeling and discretization. The continuous POD and the fundamental idea behind it will be reviewed in Section 9.3. The design of a sliding mode controller and the proof of closed-loop system stability and reference tracking is given in Section 9.4. This section also provides a discussion on the observer design for state estimation of the reduced-order model needed for the sliding mode control design. The simulation results are shown in Section 9.5, and the concluding remarks are finally made in Section 9.6.

9.2 Approximation of Burgers' Equation with Finite Element Method

In the present study, we consider this nonlinear PDE model with Neumann boundary conditions to develop a reduced order, control-oriented model. To this end, we first approximate this nonlinear PDE model with a large number of ordinary differential equations (ODEs) using finite element models (FEMs), and then reduce it to the state-space form using proper orthogonal decomposition (POD) method.

Suppose that Ω denotes the spatial interval $(0, L)$, and for $T > 0$, we define $Q = (0, T) \times \Omega$. For a given velocity $w(t, x)$ and viscosity ν, the governing viscous Burgers' PDE and the initial and boundary conditions are described by

$$\frac{\partial w(t,x)}{\partial t} + w(t,x)\frac{\partial w(t,x)}{\partial x} - \nu \frac{\partial^2 w(t,x)}{\partial x^2} = f(t,x), \tag{9.1a}$$

$$\text{I.C.}: \quad w(0,x) = w_0(x), \tag{9.1b}$$

$$\text{B.C.}: \quad w_x(t,0) = u_1(t), \quad w_x(t,L) = u_2(t), \tag{9.1c}$$

where $(t,x) \in Q$, $u_1(t)$ and $u_2(t)$ are the varying boundary conditions (i.e., the system inputs) that specify the flux condition on the boundaries. The viscosity is defined as $\nu = \frac{1}{Re}$, where Re represents the Reynolds number. The function f in (9.1a) is the forcing term assumed to be square integrable in space and time. We define the Hilbert space of Lebesgue square integrable functions as $H = L^2(\Omega)$. Note that the function $f \in H$ if it satisfies

$$\int_0^T \|f(t,x)\|_H^2 \mathrm{d}t < \infty. \tag{9.2}$$

The finite element method is a powerful tool to approximate PDEs with lumped-parameter ordinary differential equation (ODE) models. An advantage of this technique is that, unlike other methods, if the PDE is time dependent, then it can be reduced to a system of ODEs which can be integrated. Having a system of linear or nonlinear ODEs can allow to represent the model in the linear or nonlinear state-space form, which would be helpful for control synthesis purposes.

FEM Representation of the Burgers' Equation

The FE modeling of Burgers' equation is based on approximating $w(t,x)$ in the space spanned by N piecewise linear basis functions $\mathcal{N}_i(x),\ i = 1,\ldots,N$ defined in [4] as

$$w(t,x) = \sum_{i=0}^{N} \mathcal{W}_i(t)\mathcal{N}_i(x), \tag{9.3}$$

where $\mathcal{W}_i(t)$ is the nodal value at the i^{th} node and time *t*, i.e., $w(t,x_i)$ [42]. The *weak solution* approach is employed here by multiplying both sides of (9.1a) by a piecewise smooth test function $\mathcal{N}_j(x)$ and integrating in the spatial variable domain [19]. As described in [4], the reduced-order model with the input vector $U(t) = [u_1(t) \ \ u_2(t)]^\top$ is eventually obtained as the following set of $N+1$ ordinary differential equations

$$M\dot{\mathcal{W}}(t) + \nu S\mathcal{W}(t) + N(\mathcal{W}(t)) - \nu LU(t) = F(t), \tag{9.4}$$

where $\mathcal{W}^2(t) = [\mathcal{W}_0^2(t) \ \dots \ \mathcal{W}_N^2(t)]^\top$ and

$$L = \begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}^\top_{(N+1)\times 2},$$

and

$$[M]_{ij} = \big\langle \mathcal{N}_i(x), \mathcal{N}_j(x) \big\rangle, \quad [S]_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j'(x) \big\rangle,$$
$$F_j(t) = \big\langle f(t,x), \mathcal{N}_j(x) \big\rangle, \tag{9.5}$$

where $(.)'$ denotes the differentiation operator, and

$$N(\mathcal{W}(t)) = \frac{1}{2} K \mathcal{W}^2(t), \tag{9.6}$$

represents the nonlinear term in the ODEs, where

$$[K]_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j(x) \big\rangle.$$

The initial condition must be specified in order to solve this set of nonlinear ODEs. To do so, the given initial condition is described in the space spanned by the basis functions, i.e.,

$$w_0(x) \approx w(0,x) = \sum_{i=0}^{N} \mathcal{W}_i(0) \mathcal{N}_i(x). \tag{9.7}$$

By multiplying the two sides of (9.7) by the test function $\mathcal{N}_j(x)$ and again employing the weak solution approach, this can be represented in the matrix form as [4]

$$M\mathcal{W}(0) = \mathcal{P}, \tag{9.8}$$

where $\mathcal{P}_j = \big\langle w_0(x), \mathcal{N}_j(x) \big\rangle$. The solution to the linear equation (9.8) gives the initial conditions needed to solve the set of ODEs in (9.4).

9.3 Proper Orthogonal Decomposition Method and its Application to Burgers' Equation

Let $Y = [y_1, \dots, y_n]_{m\times n}$ be a real-valued data matrix containing n temporal snapshot vectors of m spatial data points. The POD basis of rank l is optimal in the sense of representing the columns

of Y, i.e., $\{y_j\}_{j=1}^n$, as a linear combination of orthonormal bases of rank l [72]. The optimality is achieved by minimizing the continuous error function between the data and its projection onto the basis set $\{\psi_i\}_{i=1}^l$, $\psi_i \in \mathbb{R}^m$ [64]

$$J = \int_0^T \left\| y(t) - \sum_{i=1}^{l} \langle y(t), \psi_i \rangle_M \psi_i \right\|_M^2 \mathrm{d}t \tag{9.9}$$
$$\text{s.t.} \quad \langle \psi_i, \psi_j \rangle_M = \delta_{ij} \quad \text{for } 1 \leq i, j \leq l.$$

As described in [4], the solution to the above constrained optimization problem leads to the following eigenvalue problem

$$\mathcal{R}^n \psi_i = \lambda_i \psi_i \ \text{ for } i = 1, \ldots, l, \tag{9.10}$$

where the linear, bounded and self-adjoint operator $\mathcal{R}^n : \mathbb{R}^m \to \mathbb{R}^m$ is defined according to the optimality condition in [4].

Next, the derivation of the reduced-order model for the Burgers' equation using POD method is described. To this purpose, we use the approximation of $w(t, x)$ in the space spanned by the POD basis functions $\psi_i(x),\ i = 1, \ldots, l$, as

$$w(t, x) = \sum_{i=1}^{l} \langle w(t, x), \psi_i(x) \rangle_M \psi_i(x). \tag{9.11}$$

By setting

$$\mathcal{W}_i^l(t) = \langle w(t, x), \psi_i(x) \rangle_M, \tag{9.12}$$

we obtain the *Galerkin* projection onto the POD space, that is

$$w(t, x) = \sum_{i=1}^{l} \mathcal{W}_i^l(t) \psi_i(x), \tag{9.13}$$

where the Fourier coefficients $\mathcal{W}_i^l$, $1 \leq i \leq l$, are functions mapping $[0, T]$ onto $\mathbb{R}$. Since for $l = m$, we have $w(t, x) = w^l(t, x)$, it can be deduced that $w^l(t, x)$ gives an approximation of $w(t, x)$ provided that $l \leq m$. In our previous work [4], we showed that the weak solution approach with the piecewise smooth test function $\psi_j(x), j = 1, 2, \ldots, l$ along with the application of POD Galerkin projection would result in the following reduced-order model

$$M^l \dot{\mathcal{W}}^l(t) + \nu S^l \mathcal{W}^l(t) + N^l(\mathcal{W}^l(t)) - \nu L^l U = F^l(t), \tag{9.14}$$

where

$$[M^l]_{ij} = \big\langle \psi_i(x), \psi_j(x) \big\rangle, \quad [S^l]_{ij} = \big\langle \psi_i'(x), \psi_j'(x) \big\rangle,$$
$$F_j^l(t) = \big\langle f(t,x), \psi_j(x) \big\rangle,$$
$$(N^l(\mathcal{W}^l(t)))_j = \frac{1}{2}\int_0^L N\big(w(t,x)\big)\psi_j(x)\,\mathrm{d}x, \tag{9.15}$$

and

$$F^l(t) = (\Psi^l)^\top F(t), \quad L^l = (\Psi^l)^\top L. \tag{9.16}$$

where $\Psi^l = [\psi_1, \psi_2, \ldots, \psi_l]$. It is noted that when the basis functions are orthonormal, $M^l = I_r$ and matrix S^l in the reduced-order model can be obtained from the original full-order matrices by expanding the POD basis functions as

$$S^l = (\Psi^l)^\top S \Psi^l. \tag{9.17}$$

The POD basis functions can be written as a linear combination of the FE basis functions, and hence

$$\sum_{i=0}^{N} \mathcal{W}_i(t)\mathcal{N}_i(x) = \sum_{i=1}^{l} \mathcal{W}_i^l(t) \sum_{m=0}^{N} \Psi_{mi}\mathcal{N}_m(x). \tag{9.18}$$

This describes the Fourier coefficients in a compact form as

$$\mathcal{W}(t) = \Psi^l \mathcal{W}^l(t). \tag{9.19}$$

Substituting (9.19) into (9.15) and using Hadamard product notation, we obtain

$$N^l(\mathcal{W}^l(t)) = \frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)), \tag{9.20}$$

and hence the reduced-order model is described by

$$\dot{\mathcal{W}}^l(t) + \nu S^l \mathcal{W}^l(t) + \frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)) - \nu L^l U(t) = F^l(t). \tag{9.21}$$

STATE-SPACE REPRESENTATION OF THE REDUCED-ORDER MODELS

A state-space representation with $N+1$ states for the ODE model obtained from finite element method, i.e., (9.4), can be determined as

$$\dot{\mathcal{W}}(t) = A\mathcal{W}(t) + \mathbf{h}(t, \mathcal{W}(t), U(t)), \tag{9.22}$$

where

$$A = -\nu M^{-1}S, \mathbf{h}(t, \mathcal{W}(t), U(t)) = -\frac{1}{2}M^{-1}K\mathcal{W}^2(t) + M^{-1}F(t) + \nu M^{-1}LU(t).$$

Also, the state-space equivalent of the reduced-order model (9.21) can be represented as

$$\dot{\mathcal{W}}^l(t) = A^l\mathcal{W}^l(t) + \mathbf{g}(t, \mathcal{W}^l(t), U(t)), \tag{9.23}$$

where

$$A^l = -\nu S^l, \mathbf{g}(t, \mathcal{W}^l(t), U(t)) = -\frac{1}{2}(\Psi^l)^\top K(\Psi^l\mathcal{W}^l(t)) \circ (\Psi^l\mathcal{W}^l(t)) + F^l(t) + \nu L^l U(t).$$

9.4 SLIDING MODE CONTROL DESIGN USING THE REDUCED-ORDER MODEL

The use of a high order controller for real-time control of complex systems is not practical due to the computational complexities involved in both control design process and its implementation. Therefore, the need for a low-order controller is inevitable. The "reduce then design" approach employs the reduced-order state space model given by (9.23) with an adequately small value of l for control design purposes. In this section, we will explain in detail how a sliding mode controller is designed on the basis of the POD-based reduced order model leading to a stable closed-loop system for the original full-order model.

9.4.1 SLIDING MODE CONTROL DESIGN

To design a sliding mode controller, a sliding surface should be defined first. To this end, we refer to the state-space representation of the system in (9.22) accompanied with the full-order system

model outputs (or the measurement equations) as

$$\begin{aligned} \dot{\mathcal{W}}(t) &= A\mathcal{W}(t) + \mathbf{h}(t, \mathcal{W}(t), U(t)) \\ Y(t) &= C\mathcal{W}(t), \end{aligned} \tag{9.24}$$

where $\mathcal{W}(t)$ represents the state vector of the full-order model. Also, C represents the system measurement matrix. We assume that only the velocities on the boundaries are measurable, and hence

$$C = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{2\times(N+1)}.$$

The reduced-order model can be described in the state-space form as

$$\begin{aligned} \dot{\mathcal{W}}^l(t) &= A^l\mathcal{W}^l(t) + \mathbf{g}(t, \mathcal{W}^l(t), U(t)), \\ Y(t) &= C^l\mathcal{W}^l(t), \end{aligned} \tag{9.25}$$

where $\mathcal{W}^l(t)$ represents the state vector of the reduced-order model. The reduced measurement matrix C^l is obtained by

$$C^l = C\Psi^l. \tag{9.26}$$

The objective of the control synthesis is to stabilize the closed-loop system and guarantee a reference trajectory tracking while being robust to uncertainties. The sliding surface to ensure that the tracking is eventually achieved is defined on the full-order model as

$$S(t) = Y(t) - r(t) = C\mathcal{W}(t) - r(t), \tag{9.27}$$

where $S(t) = [S_1(t) \quad S_2(t)]^\top$, $Y(t) = [y_1(t) \quad y_2(t)]^\top$ is the system output and $r(t) = [r_1(t) \quad r_2(t)]^\top$ is the reference signal. The sliding mode controller (SMC) needs an on-line access to the reduced-order model states. Thus, a nonlinear low-order functional observer is designed using the method proposed in [5] to estimate the states of the reduced-order model. The design of the observer will be described at the end of this section. The SMC law usually includes a switching control law and an equivalent control law [112]. A switching control law is employed to drive the system states towards a predefined sliding surface while the equivalent control law

guarantees that the system states remain around the sliding surface and converge to the surface. The control law is considered as

$$U(t) = u_{eq}(t) + u_{sw}(t), \tag{9.28}$$

where $u_{eq}(t)$ and $u_{sw}(t)$ represent the equivalent control and the switching control laws, respectively. To construct the equivalent dynamics, the full-order model is considered while the states are sliding on the defined surface. In fact, only when the system states are on the surface, the equivalent control provides an action. Taking the derivative of (9.27) results in

$$\dot{S} = C\dot{\mathcal{W}}(t) - \dot{r}(t) = C\bigg(A\mathcal{W}(t) - \frac{1}{2}M^{-1}K\mathcal{W}^2(t) + M^{-1}F(t) + \nu M^{-1}Lu_{eq}(t)\bigg) - \dot{r}(t). \tag{9.29}$$

9.4.2 Stability Analysis of the Closed-loop System with the Proposed Sliding Model Controller

In order to investigate the asymptotic stability of the closed-loop system with the sliding mode control law, following Lyapunov function is considered

$$V(t) = \frac{1}{2}S^\top S, \tag{9.30}$$

where S is the sliding surface defined in (9.27). The stability of the system given by (9.24) is guaranteed for the sliding surface (9.27) if

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} < 0 \quad \text{or} \quad S^\top \dot{S} < 0 \tag{9.31}$$

in a neighborhood of the surface given by $S(\mathcal{W}) = 0$. Substituting (9.29) into (9.31) and using (9.22), we have

$$S^\top \dot{S} = \big(C\mathcal{W}(t) - r(t)\big)^\top \big(C\dot{\mathcal{W}}(t) - \dot{r}(t)\big) = \mathcal{W}^\top C^\top C\bigg(A\mathcal{W}(t) - \frac{1}{2}M^{-1}K\mathcal{W}^2(t) + M^{-1}F(t) + \nu M^{-1}LU(t)\bigg) - \mathcal{W}^\top C^\top \dot{r} - r^\top C\bigg(A\mathcal{W}(t) + A\mathcal{W}(t) - \frac{1}{2}M^{-1}K\mathcal{W}^2(t) + M^{-1}F(t) + \nu M^{-1}LU(t)\bigg) + r^\top \dot{r}. \tag{9.32}$$

As observed from (9.32), the defined surface is a function of the states of the full-order model. However, the main goal is to implement the control law obtained from the reduced-order model instead of using the full-order one. To this end, the reduced-order model will be used in the defined surface to find the equivalent control law as

$$S(t) = Y(t) - r(t) = C^l \mathcal{W}^l(t) - r(t). \tag{9.33}$$

The model dynamics while on the sliding surface can be obtained from

$$\dot{S}(t) = \dot{Y}(t) - \dot{r}(t) = C^l \dot{\mathcal{W}}^l(t) - \dot{r}(t) = 0. \tag{9.34}$$

Solving this equation for u_{eq} by substituting (9.25) into (9.34) gives

$$u_{eq} = (\nu C^l L^l)^{-1} \left[\dot{r}(t) - C^l \left(A^l \mathcal{W}^l(t) - \frac{1}{2} (\Psi^l)^\top K(\Psi^l \mathcal{W}^l(t)) \circ (\Psi^l \mathcal{W}^l(t)) + F^l(t) \right) \right]. \tag{9.35}$$

Finally, substituting (9.28) and (9.35) into (9.32) results in the following

$$\begin{aligned} \frac{\mathrm{d}V(t)}{\mathrm{d}t} = S^\top \Bigg[C \Bigg(A\mathcal{W}(t) - \frac{1}{2} M^{-1} K \mathcal{W}^2(t) + M^{-1} F(t) \Bigg) - (\nu C M^{-1} L)(\nu C^l L^l)^{-1} C^l \Bigg(A^l \mathcal{W}^l(t) \\ - \frac{1}{2} (\Psi^l)^\top K(\Psi^l \mathcal{W}^l(t)) \circ (\Psi^l \mathcal{W}^l(t)) + F^l(t) \Bigg) \Bigg] + S^\top \Big[(\nu C M^{-1} L)(\nu C^l L^l)^{-1} - I_{2\times 2} \Big] \dot{r}(t) \\ + S^\top (\nu C M^{-1} L) u_{sw} < 0. \end{aligned} \tag{9.36}$$

The latter equation can be represented in a simpler form by rewriting the reduced-order model matrices in terms of the full-order ones using (9.16) and (9.26) as

$$\nu C^l L^l = \nu C \Psi^l (\Psi^l)^\top L. \tag{9.37}$$

On the other hand, the definition of the weighted product and orthonormality of the basis functions leads to

$$M^l = (\Psi^l)^\top M \Psi^l = I_r. \tag{9.38}$$

After some matrix manipulations, we obtain the following expression for the matrix M,

$$M = \left(\Psi^l (\Psi^l)^\top \right)^{-1}. \tag{9.39}$$

Combining (9.37) and (9.39), we obtain

$$\nu C^l L^l = \nu C \Psi^l (\Psi^l)^\top L = \nu C M^{-1} L. \tag{9.40}$$

Furthermore, the reduced-order model matrices in (9.36) can be written in terms of the full-order model matrices using the equations (9.17), (9.19) and (9.26) as

$$\begin{aligned} C^l \bigg(A^l \mathcal{W}^l(t) - \frac{1}{2}(\Psi^l)^\top K(\Psi^l \mathcal{W}^l(t)) \circ (\Psi^l \mathcal{W}^l(t)) + F^l(t) \bigg) &= C\Psi^l(-\nu(\Psi^l)^\top S \Psi^l)\mathcal{W}^l(t) \\ &\quad - \frac{1}{2} C\Psi^l(\Psi^l)^\top K(\Psi^l \mathcal{W}^l(t)) \circ (\Psi^l \mathcal{W}^l(t)) + C\Psi^l(\Psi^l)^\top F(t) \\ &= C(-\nu M^{-1} S)\mathcal{W}(t) + \frac{1}{2} C M^{-1} K \mathcal{W}(t) \circ \mathcal{W}(t) + C M^{-1} F(t) \\ &= C\bigg(A\mathcal{W}(t) - \frac{1}{2} M^{-1} K \mathcal{W}^2(t) + M^{-1} F(t) \bigg). \end{aligned} \tag{9.41}$$

By substituting (9.41) back into (9.36), we have

$$S^\top (\nu C M^{-1} L) u_{sw} < 0, \tag{9.42}$$

which implies that the switching control law only needs to satisfy the inequality condition (9.42). Considering (9.40) and the fact that the system under study is a multi-input multi-output system, the switching control law corresponding to the defined surfaces can be written as

$$u_{sw}(t) = -(\nu C^l L^l)^{-1} \begin{bmatrix} \lambda_1 S_1 + \xi_1 sat(S_1) \\ \lambda_2 S_2 + \xi_2 sat(S_2) \end{bmatrix}, \tag{9.43}$$

where λ_1, λ_2, ξ_1 and ξ_2 are positive constants and $sat(.)$ is the saturation function with the upper limit of 1 and lower limit of -1. These constants, which are chosen by trial and error considering the trade-off between the reaching time and chattering, can be considered large enough when the trajectory is far from the switching surface (so that the reaching time is short), and then as small as desired in order to limit the chattering.

9.4.3 Functional Observer Design

A functional observer can be designed to estimate the states of the reduced-order model instead of the full-order one. Hence, the computational cost would be much lower compared to the full-order state observer. The reduced-order model described by (9.23) is obtained from the discretized

Burgers' equation with a locally Lipschitz nonlinearity with respect to $\mathcal{W}^l$ in a region $\mathcal{D}$, i.e., for any $\mathcal{W}_1^l(t), \mathcal{W}_2^l(t) \in \mathcal{D}$ [5]

$$\|\mathbf{g}(\mathcal{W}_1^l, U^*) - \mathbf{g}(\mathcal{W}_2^l, U^*)\| \leq \gamma_d \|\mathcal{W}_1^l - \mathcal{W}_2^l\|, \tag{9.44}$$

where $\| \ . \ \|$ represents the induced 2-norm, U^* is an admissible control sequence and γ_d is the nonnegative Lipschitz constant. If the nonlinear function $\mathbf{g}(.,.)$ globally satisfies the Lipschitz continuity condition in $\mathbb{R}^l$, then the global stability of the observer is guaranteed [5]. The proposed observer takes the following form

$$\dot{\hat{\mathcal{W}}}^l(t) = A^l \hat{\mathcal{W}}^l(t) + \mathbf{g}(\hat{\mathcal{W}}^l(t), U(t)) + L(Y(t) - C^l \hat{\mathcal{W}}^l(t)), \tag{9.45}$$

where $\hat{\mathcal{W}}^l$ is the estimate of $\mathcal{W}^l$ and L is the observer matrix gain chosen such that the observer error system is asymptotically stable. As shown in [5], a sufficient linear matrix inequality (LMI) condition can be determined to maximize γ_d while a stabilizing L is obtained. The observer of order l in (9.45) is used to estimate $\mathcal{W}^l$ considering that the quadratic nonlinearities in the reduced-order model described by (9.23) are locally Lipschitz.

9.5 Simulation Results and Discussion

In this section, we illustrate some of the results of our numerical studies and further provide a discussion on the accuracy of the derived reduced-order models, as well as the high performance of the closed-loop system achieved from the implementation of the designed sliding mode controller.

9.5.1 Open-loop System Simulation Results to Examine the Reduced-oder Model Accuracy

In order to assess the performance of the presented model reduction method, an example of a viscous Burgers' equation is examined here. The forcing term in (9.1a) is considered to be zero,

which translates to the so-called viscous Burgers' equation. The initial condition is assumed to be

$$w_0(x) = \begin{cases} 100(sin(8\pi x) - 2x), & \text{for } x \in (0, \frac{1}{4}] \\ 0, & \text{otherwise.} \end{cases}$$

It is also assumed that the boundary conditions, i.e., inputs to the state-space models, are sinusoidal functions as

$$u_1(t) = 0.8sin(2t), \quad u_2(t) = 0.5sin(t). \tag{9.46}$$

The reduced and full-order open-loop models are simulated for a given viscosity $\nu = 0.01$ (or *Re* = 100) to gauge the performance of the model reduction approach for a class of physical flows. It is observed that by increasing the number of basis functions to 7, a very close match between the outputs of the two models can be achieved (see [4]). Finally, to quantify the model accuracy, we consider the *Best Fit Rate* (BFR) as implemented in [4]. Table 9.1 shows the BFR of the reduced-order models. Also shown in Table 9.1 is the mean-squared error (MSE) between the reduced-order model output and the original FE model output.

Table 9.1: The MSE and BFR of the output signal of the reduced order models with sinusoidal inputs (9.46)

	POD (3 Bases)		POD (5 Bases)		POD (7 Bases)	
Output	MSE	BFR	MSE	BFR	MSE	BFR
y_1	0.0021	0.0886	2.1836e-04	0.7042	1.3275e-05	0.9271
y_2	1.0203e-04	0.4909	6.7584e-06	0.8690	4.2925e-06	0.8956

9.5.2 Closed-loop Simulation Results with the Designed Sliding Mode Controller

The discretized full-order model obtained from FEM is used to validate the designed sliding mode controller (SMC) consisting of equivalent and switching control laws in tracking a given reference trajectory. From the fluid mechanics point of view, this can be seen as the problem of controlling the flux on the boundaries to reach the desired flow velocity at the desired points. The viscous

Burgers' equation with the same initial condition as given in the previous section is used to validate the proposed SMC design approach. The two components of the SMC laws are obtained from the 7^{th} order reduced model. As described before, the measurement devices are considered to be placed on the boundaries to collect the flow velocity as the system output. A sinusoidal signal is considered as the reference input and the system output and control input are shown for the reference input. The tracking performance and control inputs for a given sinusoidal reference

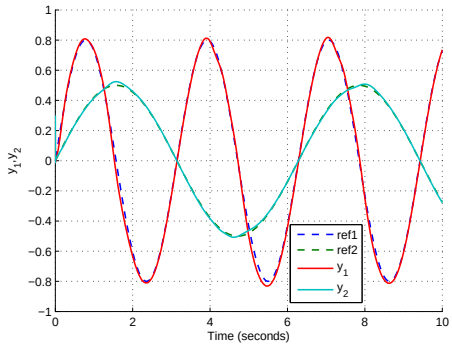


Figure 9.1: System outputs and the reference inputs for the given sinusoidal function.

signal are shown in Figures 9.1 and 9.2. The corresponding sliding mode controller parameters are tuned as $\lambda_1 = 1.3\times10^4,\ \lambda_2 = 1.95\times10^4,\ \xi_1 = 1.4\times10^3$ and $\xi_2 = 1.3\times10^3$. The simulation result of the full-order model is shown in Figure 9.3. As observed from Figure 9.1, the proposed SMC law illustrates a high tracking performance in the presence of model uncertainties. Uncertainties in the problem in hand are primarily due to the discrepancy between the full-order and reduced-order models, where the number of the eigenfunctions chosen to find the POD bases dictates the level of uncertainties. The switching control in the SMC law works in favor of keeping the trajectory on the defined sliding surface in the presence of the aforedescribed uncertainties.

9.6 Concluding Remarks

In this paper, the developed reduced-order model in [4] has been used for the design of a sliding mode controller on the basis of the sliding surfaces defined according to the reduced-order model. Due to the need for state estimates required by the sliding mode controller, implementing

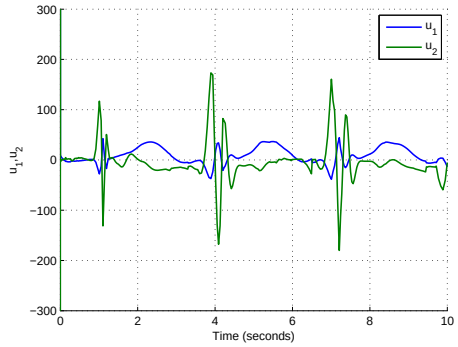


Figure 9.2: The control inputs from the proposed SMC law to track a sinusoidal function.

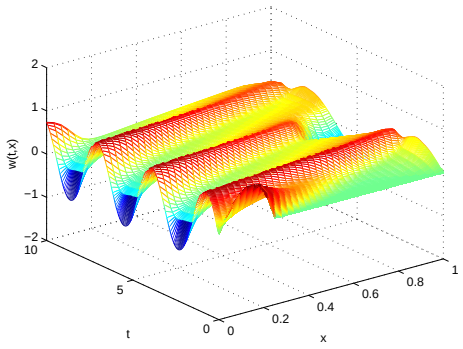


Figure 9.3: Velocity response of the full-order model with the proposed SMC law to track a sinusoidal function.

the reduced-order model significantly decreased the computational load for both controller and observer design. Finally, numerical studies have demonstrated promising results by using the proposed reduced-order model and controller to achieve a high performance tracking of different reference trajectories. This overall proved the practicality of the proposed control-oriented modeling and model-based nonlinear control design approach for complex systems governed by nonlinear PDEs.

CHAPTER 10

ROBUST NONLINEAR CONTROL DESIGN FOR SYSTEMS GOVERNED BY BURGERS' EQUATION SUBJECTED TO PARAMETER VARIATION [1]

[1] F. Abbasi, J. Mohammadpour: Robust Nonlinear Control Design for Systems Governed by Burgers' Equation Subjected to Parameter Variation. 2016. *Submitted to International Journal of Control*.

ABSTRACT

In this paper, a robust sliding mode controller is proposed for dynamic systems governed by Burgers' equation with Neumann boundary conditions in the presence of parameter variations. The main objective is to design a reduced-order model based controller at a nominal value of the system parameter that stabilizes the full order model while being robust with respect to model uncertainties in the obtained reduced-order model. The model uncertainties resulted from the variation of the parameter ν are discussed under two categories; first, the error arising from the change in the states of the full-order model, and second, the error associated with the estimated proper orthogonal decomposition (POD) basis functions at the nominal value of ν_0. In the present work, the boundedness of the error functions is studied and an estimation of these bounds is obtained in terms of the reduced-order model and matrices of the full-order system that are known *a priori*. Next, the bounds on the error functions is used to design a reduced-order sliding mode controller that guarantees the stability of the full-order model obtained via a finite element approximation of the Burgers' equation for a trajectory tracking problem.

10.1 INTRODUCTION

Computational modeling, simulation and control of nonlinear turbulent systems is a challenging task due to the complexity of the fluid mechanics problems. There are handful of researches targeting the control problem of linear, semi-linear and nonlinear parabolic and hyperbolic partial differential equations (PDEs). In [52], stabilization of a semi-linear parabolic partial differential equation, in which the heat source depends on the temperature of the whole space, is considered by using boundary control. The adaptive boundary stabilization and control has been investigated for a class of systems described by first-order hyperbolic PDEs with unknown spatially varying parameter in [153]. Also, the predictive control of linear parabolic partial differential equations (PDEs) with state and control constraints was studied in [38]. The Navier-Stokes equation describes many of the underlying phenomena in fluid mechanics. This equation is simplified to Burgers' equation

when flow is considered to be incompressible and with the pressure term removed. The Burgers' equation can also be viewed as an intermediate step to capture very critical nonlinear convective behaviors that can model shock waves, some boundary layer problems and traffic flow problems among many others [6, 20].

There are a handful of discretization methods that can be employed to solve complex nonlinear partial differential equations like Burgers' equation [59]. However, the implementation of the standard discretization schemes such as finite element or finite difference methods may require a large number of degrees of freedom to accurately describe the fluid flows. Additionally, the control design task based on the full-order model would be cumbersome. However, with respect to both storage and computational time, these methods are inefficient. This can also be crucial when the real-time solutions for feedback control of complex systems are sought [15]. As a remedy, the reduced-order modeling was introduced to approximate the original mathematical model by a simpler one so it could still represent certain significant aspects and dominant dynamics of the system with an acceptable accuracy depending on the complexity of the reduced-order model [18, 73].

The use of reduced-order models can introduce a source of uncertainty imposed by the order of the reduced model. In addition to this, there are other types of uncertainty, i.e., varying parameter, that might affect the accuracy of the extracted reduced model [16, 57, 153]. Hence, the objective is to obtain an accurate reduced-order representation of the original system while ensuring the robustness to uncertainties. An approach based on a dictionary of solutions is developed by [6] as an alternative to using a truncated reduced basis based on proper orthogonal decomposition. The elements of this dictionary are solutions computed for varying values of time and the associated parameter. In [56], a sensitivity analysis is carried out to include the flow and shape parameters influenced during the basis selection process to develop more robust reduced order models for varying viscosity, changing orientation and shape definition of bodies.

The proper orthogonal decomposition (POD), also known as the Karhunen-Loeve decomposition, has proven to be an efficient tool for model reduction [72, 90]. POD extracts a number of basis functions that would be used in a collocation formulation of Galerkin projection resulting in a finite

dimensional system with a low number of degrees of freedom. The generated POD basis functions solving the underlying eigenvalue problem are dependent on a set of parameters may not give an accurate estimation of the full-order model associated with a different set of parameters [9,58]. The POD bases must be regenerated for each set of parameters to determine an accurate reduced-order estimation of the perturbed problem which is a computationally inefficient process [8].

In the present paper, the model uncertainties are classified in two categories and a robust nonlinear controller is proposed for a trajectory tracking problem. First, the model uncertainties arising from the approximation of the full-order model by the reduced-order one are investigated. Furthermore, the error associated with the varying parameter is studied when the POD basis functions extracted at the nominal parameter ν_0 are used to estimate the full-order model at the new value of the viscosity ν. A nonlinear control strategy based on the reduced-order sliding mode control is then proposed to tackle different kinds of uncertainties arising from parametric and modeling imprecisions in the reduced-order nonlinear model of Burgers' equation. Sliding mode control (SMC) is a nonlinear feedback control scheme that can effectively apply a high-frequency switching control to alter the dynamics of a nonlinear system [127]. Switching from one continuous mode to another considering the system's current position in state space can guarantee the convergence of the trajectories towards a switching surface that eventually slides along the boundaries of the control structure.

The advantages of the presented approach are twofold. First, the obtained reduced model is calculated once and the model uncertainties are bounded with the reduced and full-order models in the nominal viscosity ν_0. In other words, instead of cumbersome calculation of the basis functions associated with different values of the parameter, the reduced-order model is computed once and an accurate estimation of the associated model uncertainties is obtained. Second, a robust controller is designed taking into account the bounds on uncertainties to capture uncertain reduced model at the nominal parameter to ensure the desired tracking of the reference trajectory for the full-order model at any viscosity around ν_0. The results of this paper demonstrate that the reduced-order model based sliding mode controller can effectively control the full-order model in the presence of

modeling uncertainties. This can significantly decrease the computational load for real-time control of the large order model of Burgers' equation.

Throughout this paper, unless otherwise specified, notation $\langle ., . \rangle$ represents the inner product of the given basis functions in the finite element method, which is the spatial-domain integration of the product of the given basis functions. Also, $\mathcal{W}_i^l$ represents the i^{th} Fourier coefficient of the reduced-order model of order l and $\mathbb{R}^m$ is the m-dimensional Euclidean space. Moreover, Kronecker delta, δ_{ij}, returns zero for $i \neq j$ and 1 for $i = j$. Finally, we define $A \circ B$ as the Hadamard product of the matrices A and B of the same dimension such that $[A \circ B]_{ij} = [A]_{ij}[B]_{ij}$.

The rest of the paper is organized as follows. Section II describes the Burgers' PDE and its finite element modeling along with the continuous POD and the extracted reduced-order model. This section also discusses the process of obtaining reduced-order model of Burgers' equation using POD. The error estimates and sliding mode control design using reduced-order model for a reference tracking problem is introduced in Section III. The simulation results are shown in Section IV, and the concluding remarks are finally made in Section V.

10.2 Approximation of Burgers' Equation with Finite Element and Proper Orthogonal Methods

Over the past three decades, Burgers' equation has been used to gain a better understanding of turbulence and other complex phenomena in fluid systems. This nonlinear parabolic partial differential equation (PDE) provides a mathematical model that can be used for boundary control and distributed feedback control design purposes. In the present study, we consider this nonlinear PDE with Neumann boundary conditions to develop a reduced order, control-oriented model. To this end, we first approximate this nonlinear PDE with a large number of ordinary differential equations (ODEs) using finite element method (FEM), and then reduce it to the state-space form using proper orthogonal decomposition (POD) method.

Suppose that Ω represents the spatial interval $(0, L)$ and that for $T > 0$, we define $Q = (0, T) \times \Omega$. Denoting the fluid velocity by $w(t, x; \nu)$ that is dependent on the viscosity ν as the

characteristic parameter, the governing viscous Burgers' PDE and the initial and boundary conditions are described by

$$\frac{\partial w_\nu(t,x)}{\partial t} + w_\nu(t,x)\frac{\partial w_\nu(t,x)}{\partial x} - \nu\frac{\partial^2 w_\nu(t,x)}{\partial x^2} = f(t,x), \tag{10.1a}$$

$$\text{I.C. :} \quad w_\nu(0,x) = w_0(x), \tag{10.1b}$$

$$\text{B.C. :} \quad w_{\nu,x}(t,0) = u_1(t), \quad w_{\nu,x}(t,L) = u_2(t), \tag{10.1c}$$

where $(t,x) \in Q$, $u_1(t)$ and $u_2(t)$ are the varying boundary conditions (i.e., controlled inputs) that specify the flux condition on the boundaries. Also, $w_{\nu,x}$ represents the spatial derivative of w_ν. The viscosity is defined as $\nu = \frac{1}{Re}$, where Re denotes the Reynolds' number. The function f in (10.1a) is the force term assumed to be square integrable in space and time. We define the Hilbert space of Lebesgue square integrable functions as $H = L^2(\Omega)$. The function f is said to be in H if it satisfies

$$\int_0^T \left\| f(t,x) \right\|_H^2 \mathrm{d}t < \infty.$$

10.2.1 State-space representation of Burgers' PDE via finite element method

The finite element method (FEM) is a powerful tool to approximate PDEs with lumped parameter ordinary differential equations (ODEs). An advantage of this method over other methods is that if the PDE is time dependent, then it can be reduced to a system of ODEs and then integrated. Having a system of linear or nonlinear ODEs can allow to represent the model in the linear or nonlinear state-space form, which would be helpful for control synthesis purposes.

if we Let $V = H^1(\Omega)$ be the associated Sobolev space as introduced in [43] and define the set of square integrable functions belonging to the associated Banach space as $v \in L^2(0,T;V)$ and $v_t \in L^2(0,T;V)$, where v_t represents the time derivative of v. Furthermore, assuming that the given initial condition $w_0(x)$ and the forcing term $f(t,x)$ belong to the space of essentially bounded functions, i.e., $w_0(x) \in L^\infty(\Omega)$ and $f(t,x) \in L^\infty(Q)$, we introduce the Banach space $\mathcal{P} = L^2(0,T;V) \bigcap L^\infty(Q)$. Hence, there exists a unique weak solution to (10.1) such that $w(t,x) \in \mathcal{P}$.

The *weak solution* approach is employed here by multiplying both sides of (10.1a) by a piecewise smooth test function $v(x)$ and integrating in the spatial variable domain [51]. Taking the integral from both sides and substituting the second order derivative term by its equivalent obtained by using the chain rule results in

$$\int_0^L \left(w_{\nu,t}(t,x) + \frac{1}{2}[w_\nu^2(t,x)]_x \right) v(x)\,\mathrm{d}x - \nu \left[u_2(t)v(L) - u_1(t)v(0) - \int_0^L w_{\nu,x}(t,x)v'(x)\,\mathrm{d}x \right] = \int_0^L f(t,x)v(x)\,\mathrm{d}x, \quad (10.2)$$

where $v'(x) = \frac{dv}{dx}$ and $w_{\nu,t} = \frac{\partial w_\nu}{\partial t}$.

In order to discretize in the spatial domain, the spatial variable interval is divided into N subintervals $[x_j, x_{j+1}]$. We define $h_j = x_{j+1} - x_j$ and assume that all the elements, denoted by e, are of equal size (uniformly spaced mesh) and hence $h_1 = \ldots = h_N = h$. The FEM basis functions are hence defined as follows [84]

(i) for element e_0,

$$\mathcal{N}_0(x) = \begin{cases} \frac{x_1 - x}{h}, & 0 \le x \le x_1 \\ 0, & \text{otherwise} \end{cases}$$

(ii) for element $e_j,\ j = 1 : N-1$,

$$\mathcal{N}_j(x) = \begin{cases} \frac{x - x_{j-1}}{h}, & x_{j-1} \le x \le x_j \\ \frac{x_{j+1} - x}{h}, & x_j \le x \le x_{j+1} \\ 0, & \text{otherwise} \end{cases}$$

(iii) for element e_N

$$\mathcal{N}_N(x) = \begin{cases} \frac{x - x_{N-1}}{h}, & x_{N-1} \le x \le x_N \\ 0, & \text{otherwise.} \end{cases}$$

The approximation of $w_\nu(t,x)$ in the space spanned by the piecewise linear basis functions is given by

$$w_\nu(t,x) = \sum_{i=0}^{N} \mathcal{W}_{\nu,i}(t)\mathcal{N}_i(x), \quad (10.3)$$

where $\mathcal{W}_{\nu,i}(t)$ is the nodal value at the i^{th} node and time *t*, i.e., $w_\nu(t, x_i)$.

Lemma 10.2.1 *Assume that there exists a weak solution to (10.1). Then, a state-space representation of (10.1) enforcing initial and Neumann boundary conditions for the given input vector* $U(t) = [u_1(t) \;\; u_2(t)]^\top$ *and Fourier coefficients* $\mathcal{W}_\nu(t) = [\mathcal{W}_{\nu,0}(t) \;\; \dots \;\; \mathcal{W}_{\nu,N}(t)]^\top$ *is determined to be*

$$\dot{\mathcal{W}}_\nu(t) = M^{-1}A\mathcal{W}_\nu(t) + M^{-1}\boldsymbol{h}(t, \mathcal{W}_\nu(t), U(t)), \tag{10.4}$$

where

$$A = -\nu S, \quad \boldsymbol{h}_\nu(t, \mathcal{W}_\nu(t), U(t)) = -\frac{1}{2}K(\mathcal{W}_\nu(t) \circ \mathcal{W}_\nu(t)) + F(t) + \nu L U(t),$$

with

$$\begin{gathered} L = \begin{bmatrix} -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}^\top_{(N+1)\times 2}, \\ [M]_{ij} = \big\langle \mathcal{N}_i(x), \mathcal{N}_j(x) \big\rangle, \; [S]_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j'(x) \big\rangle, \\ F_j(t) = \big\langle f(t,x), \mathcal{N}_j(x) \big\rangle, \; [K]_{ij} = \big\langle \mathcal{N}_i'(x), \mathcal{N}_j(x) \big\rangle, \end{gathered} \tag{10.5}$$

and with the initial condition represented in the matrix form as

$$M\mathcal{W}_\nu(0) = \mathcal{I}, \tag{10.6}$$

where $\mathcal{I}_j = \big\langle w_0(x), \mathcal{N}_j(x) \big\rangle$.

Proof 6 *Since the given Galerkin approximation of* $w_\nu(t, x)$*, i.e., equation (10.3), belongs to the Banach space of the weak solutions* $L^2(0, T; V) \bigcap L^\infty(Q)$*, hence, it can be substituted in equation (10.2). Also, the arbitrary and piecewise smooth function* $v(x)$ *is substituted by the piecewise linear basis function of the FEM,* $\mathcal{N}_j(x)$*, for* $j = 0, 1, \dots, N$*. To cope with the nonlinear term, the group finite element (GFE) method is used, which is an alternative approach for to FEM for solving nonlinear elliptic, parabolic, and hyperbolic problems [130, 152]. The nonlinear term is represented using GFE as*

$$w_\nu^2(t, x) = \sum_{i=0}^{N} \mathcal{W}_{\nu,i}^2(t)\mathcal{N}_i(x), \tag{10.7}$$

$$(\mathscr{N}(\mathcal{W}_\nu(t)))_j = \frac{1}{2}\sum_{i=0}^{N} \mathcal{W}_{\nu,i}^2(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j(x).$$

Defining

$$[K]_{ij} = \left\langle \mathcal{N}_i'(x), \mathcal{N}_j(x) \right\rangle$$

yields the following quadratic form

$$\mathscr{N}(\mathcal{W}_\nu(t)) = \frac{1}{2} K(\mathcal{W}_\nu(t) \circ \mathcal{W}_\nu(t)). \tag{10.8}$$

Using Galerkin method and substituting (10.7) into (10.2) results in

$$\sum_{i=0}^{N} \dot{\mathcal{W}}_{\nu,i}(t) \int_0^L \mathcal{N}_i(x)\mathcal{N}_j(x)\,\mathrm{d}x + \frac{1}{2}\sum_{i=0}^{N} \mathcal{W}_{\nu,i}^2(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j(x)\,\mathrm{d}x - \nu\Big[u_2(t)\mathcal{N}_j(L) - u_1(t)\mathcal{N}_j(0)\Big] + \nu \sum_{i=0}^{N} \mathcal{W}_{\nu,i}(t) \int_0^L \mathcal{N}_i'(x)\mathcal{N}_j'(x)\,\mathrm{d}x = \int_0^L f(t,x)\mathcal{N}_j(x)\,\mathrm{d}x.$$

We consider the notations defined in (10.5) to form the matrix representation of the previous equation

$$M\dot{\mathcal{W}}_\nu(t) + \nu S\mathcal{W}_\nu(t) + \frac{1}{2}K(\mathcal{W}_\nu(t) \circ \mathcal{W}_\nu(t)) - \nu LU(t) = F(t). \tag{10.9}$$

Rewriting this equation results in the space-space representation (10.4). The given initial condition (10.1b) is described in the space spanned by the basis functions as

$$w_\nu(0,x) = \sum_{i=0}^{N} \mathcal{W}_{\nu,i}(0)\mathcal{N}_i(x). \tag{10.10}$$

By enforcing the initial condition (10.1b) and multiplying it by the test function $\mathcal{N}_j(x)$ from both sides and again employing the weak solution approach, we obtain

$$\sum_{i=0}^{N} \mathcal{W}_{\nu,i}(0) \int_0^L \mathcal{N}_i(x)\mathcal{N}_j(x) = \int_0^L w_0(x)\mathcal{N}_j(x), \quad j = 0, \ldots, N.$$

This can be represented in the matrix form as (10.6). The solution to this linear equation gives the initial conditions required for the set of ODEs in (10.9).

10.2.2 REDUCED-ORDER MODEL OF BURGERS' PDE USING PROPER ORTHOGONAL DECOMPOSITION

The fundamental idea behind proper orthogonal decomposition (POD) is to optimally represent a given data set in a mean-squared error sense using an orthonormal basis of rank l. Let $\mathscr{W}_\nu = [\mathscr{W}_{\nu,1}, \ldots, \mathscr{W}_{\nu,n}]_{(N+1)\times n}$ be a real-valued data matrix containing n snapshot vectors of $N+1$ spatial data points obtained for parameter ν, where for j^{th} time snapshot t_j, we have $\mathscr{W}_{\nu,j} = [\mathcal{W}_{\nu,0}(t_j), \mathcal{W}_{\nu,1}(t_j), \ldots, \mathcal{W}_{\nu,N+1}(t_j)]^\top$. The POD basis of rank l is optimal in the sense of representing the columns of $\mathscr{W}$, i.e., $\{\mathscr{W}_j\}_{j=1}^n$, as a linear combination of orthonormal bases of rank l [72]. We endow the Euclidean space $\mathbb{R}^{N+1}$ with the weighted inner product defined as

$$\begin{aligned}\langle \psi, \tilde{\psi} \rangle_W &= \psi^T W \tilde{\psi} = \langle \psi, W\tilde{\psi} \rangle_{\mathbb{R}^{N+1}} \\ &= \langle W\psi, \tilde{\psi} \rangle_{\mathbb{R}^{N+1}} \quad \text{for } \psi, \tilde{\psi} \in \mathbb{R}^{N+1},\end{aligned}$$

where $W \in \mathbb{R}^{(N+1)\times(N+1)}$ is a symmetric, positive-definite matrix. Note that the vector $\mathscr{W}(t)$, $t \in [0, T]$, now represents a function in Ω evaluated at m grid points. Therefore, we should supply $\mathbb{R}^{N+1}$ with a weighted inner product representing a discretized inner product in an appropriate function space. Since the mass matrix in (10.9) is symmetric, real and positive definite, it can be considered as the weight matrix (i.e., W) in the aforedescribed inner product. The goal is to determine a POD basis of rank $l < n$ that gives the best estimate of the entire trajectory $\varrho^{\mathscr{W}} = \text{span}\{\mathscr{W}_\nu(t) | t \in [0, T]\} \subset \mathbb{R}^{N+1}$. The optimality is achieved by minimizing the continuous error function between the data and its projection onto the basis set $\{\psi_{\nu,i}\}_{i=1}^l$, $\psi_i \in \mathbb{R}^{N+1}$ that is obtained for a parameter value ν

$$J = \int_0^T \left\| \mathscr{W}_\nu(t) - \sum_{i=1}^{l} \langle \mathscr{W}_\nu(t), \psi_{\nu,i} \rangle_M \psi_{\nu,i} \right\|_M^2 \mathrm{d}t \tag{10.11}$$

$$\text{s.t.} \quad \langle \psi_{\nu,i}, \psi_{\nu,j} \rangle_M = \delta_{ij} \quad \text{for } 1 \le i, j \le l,$$

where projection operator associated with the space spanned by $\{\psi_{\nu,i}\}_{i=1}^l$ can be defined as

$$\mathcal{P}_{\psi_\nu}^l \mathscr{W}_\nu(t) = \sum_{i=1}^{l} \langle \mathscr{W}_\nu(t), \psi_{\nu,i} \rangle_M \psi_{\nu,i}. \tag{10.12}$$

Next, let $\bar{Y} = M^{1/2} Y D^{1/2} \in \mathbb{R}^{(N+1)\times n}$. Then, the solution to the optimization problem (10.11) can be obtained by solving the following eigenvalue problem

$$\begin{aligned} \bar{Y}\bar{Y}^T \bar{\psi}_{\nu,i} &= \lambda_{\nu,i} \bar{\psi}_{\nu,i}, \quad 1 \le i \le l, \\ \langle \bar{\psi}_{\nu,i}, \bar{\psi}_{\nu,j} \rangle_{\mathbb{R}^{N+1}} &= \delta_{ij}, \quad 1 \le i, j \le l, \end{aligned} \tag{10.13}$$

where $D = \text{diag}(\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^{n\times n}$ and α_j's denote the non-negative trapezoidal weights defined by

$$\alpha_1 = \frac{\Delta t}{2}, \ \alpha_j = \Delta t \quad \text{for } 2 \le j \le n-1, \ \alpha_n = \frac{\Delta t}{2}, \tag{10.14}$$

with $\Delta t = t_j - t_{j-1}$ [106].

The choice for the number of basis functions l, that can lead to an accurate reduced-order representation of the original model, is certainly of critical importance when applying POD. There is no general rule for selecting l; it is rather heuristic and based on the captured relative energy defined by [76]

$$\mathcal{E}(l) = \frac{\sum_{i=1}^{l} \lambda_{\nu,i}}{\sum_{i=1}^{d} \lambda_{\nu,i}}, \tag{10.15}$$

where $d = rank(\bar{Y})$.

Next, the derivation of the reduced-order model for the Burgers' equation using POD method is described. To this purpose, we use the approximation of $w_\nu(t,x)$ in the space spanned by the POD basis functions $\psi_{\nu,i}(x), \ i = 1, \ldots, l$, as

$$w_\nu(t,x) = \sum_{i=1}^{l} \langle w_\nu(t,x), \psi_{\nu,i}(x) \rangle_M \psi_{\nu,i}(x).$$

By setting

$$\mathcal{W}_{\nu,i}^l(t) = \langle w_\nu(t,x), \psi_{\nu,i}(x) \rangle_M,$$

we obtain the Galerkin form of the projection onto the POD space to approximate $w_\nu(t,x)$ as

$$w_\nu(t,x) = \sum_{i=1}^{l} \mathcal{W}_{\nu,i}^l(t) \psi_{\nu,i}(x), \tag{10.16}$$

where the Fourier coefficients $\mathcal{W}_i^l$, $1 \le i \le l$, are functions mapping $[0,T]$ onto $\mathbb{R}$.

Lemma 10.2.2 *The state-space representation of the reduced-order model satisfying the given initial condition and Neumann boundary conditions is determined as*

$$\dot{\mathcal{W}}_\nu^l(t) = A_\nu^l \mathcal{W}_\nu^l(t) + \boldsymbol{g}_\nu(t, \mathcal{W}_\nu^l(t), U(t)), \tag{10.17}$$

where

$$A_\nu^l = -\nu S_\nu^l, \tag{10.18}$$

$$\boldsymbol{g}_\nu(t, \mathcal{W}_\nu^l(t), U(t)) = (\Psi_\nu^l)^\top \boldsymbol{h}_\nu(t, \Psi_\nu^l \mathcal{W}_\nu^l(t), U(t)) = \\ -\frac{1}{2}(\Psi_\nu^l)^\top K(\Psi_\nu^l \mathcal{W}_\nu^l(t)) \circ (\Psi_\nu^l \mathcal{W}_\nu^l(t)) + F_\nu^l(t) + \nu L_\nu^l U(t),$$

with

$$M_\nu^l = I_r, \quad F_\nu^l(t) = (\Psi_\nu^l)^\top F(t), \quad L_\nu^l = (\Psi_\nu^l)^\top L, \tag{10.19}$$

$$S_\nu^l = (\Psi_\nu^l)^\top S \Psi_\nu^l, \quad N_\nu^l(\mathcal{W}_\nu^l(t)) = \frac{1}{2}(\Psi_\nu^l)^\top K(\Psi_\nu^l \mathcal{W}_\nu^l(t)) \circ (\Psi_\nu^l \mathcal{W}_\nu^l(t)). \tag{10.20}$$

Proof 7 *We recall the weak solution approach described earlier for Burgers' equation that led to*

$$\int_0^L \left(w_{\nu,t}(t,x) + \frac{1}{2}[w_\nu^2(t,x)]_x \right) v(x)\,\mathrm{d}x - \nu \left[u_2(t)v(L) - u_1(t)v(0) - \int_0^L w_{\nu,x}(t,x)v'(x)\,\mathrm{d}x \right] = \\ \int_0^L f(t,x)v(x)\,\mathrm{d}x.$$

Since $v(x)$ is arbitrary and piecewise smooth, we now choose $v(x) = \psi_{\nu,j}(x), j = 1, 2, \ldots, l$, and use POD Galerkin projection that results in

$$\sum_{i=1}^{l} \dot{\mathcal{W}}_{\nu,i}^l(t) \int_0^L \psi_{\nu,i}(x)\psi_{\nu,j}(x)\,\mathrm{d}x + \frac{1}{2}\int_0^L \left(\left[\sum_{i=1}^{l} \mathcal{W}_{\nu,i}^l(t)\psi_{\nu,i}(x) \right]^2 \right)_x \psi_{\nu,j}(x)\,\mathrm{d}x - \\ \nu \left[u_2(t)\psi_{\nu,j}(L) - u_1(t)\psi_{\nu,j}(0) \right] + \nu \sum_{i=1}^{l} \mathcal{W}_{\nu,i}^l(t) \int_0^L \psi'_{\nu,i}(x)\psi'_{\nu,j}(x)\,\mathrm{d}x = \int_0^L f(t,x)\psi_{\nu,j}(x)\,\mathrm{d}x.$$

Considering the notations introduced in (10.19) and (10.20), it is straightforward to show that

$$[M_\nu^l]_{ij} = \big\langle \psi_{\nu,i}(x), \psi_{\nu,j}(x) \big\rangle, \quad [S_\nu^l]_{ij} = \big\langle \psi'_{\nu,i}(x), \psi'_{\nu,j}(x) \big\rangle,$$

$$F_{\nu,j}^{\ l}(t) = \big\langle f(t,x), \psi_{\nu,j}(x) \big\rangle,$$

$$(N_\nu^l(\mathcal{W}_\nu^l(t)))_j = \frac{1}{2}\int_0^L \left(\left[\sum_{i=1}^{l} \mathcal{W}_{\nu,i}^l(t)\psi_{\nu,i}(x) \right]^2 \right)_x \psi_{\nu,j}(x)\,\mathrm{d}x,$$

where $w_\nu(t,x)$ is approximated by (10.16). According to the orthogonality property of the basis functions, we have $M_\nu^l = I_r$. On the other hand, the definition of the weighted product and orthonormality of the basis functions leads to

$$M^l = (\Psi^l)^\top M \Psi^l = I_r.$$

Also, matrix S_ν^l in the reduced-order model can be obtained from the original full-order matrices by expanding the POD basis functions as

$$\begin{aligned}
[S_\nu^l]_{ij} = \big\langle \psi'_{\nu,i}(x), \psi'_{\nu,j}(x) \big\rangle &= \int_0^L \psi'_{\nu,i}(x)\psi'_j(x;\nu)\,\mathrm{d}x \\
&= \int_0^L \sum_{k=0}^{N} \Psi_{ki}^\nu \mathcal{N}'_k(x) \sum_{m=0}^{N} \Psi_{mj}^\nu \mathcal{N}'_m(x)\,\mathrm{d}x \\
&= \sum_{k=0}^{N}\sum_{m=0}^{N} \Psi_{ki}^\nu \Psi_{mj}^\nu \int_0^L \mathcal{N}'_k(x)\mathcal{N}'_m(x)\,\mathrm{d}x,
\end{aligned}$$

which can be represented in the matrix form (10.20). Hence, the matrix S_ν^l has been characterized in terms of the full-order matrix S and the POD basis functions. Same procedure can be followed to represent the nonlinear term in the reduced-order model in terms of the full-order matrices. The Fourier coefficients are described in a compact form as

$$\tilde{\mathcal{W}}_\nu(t) = \Psi_\nu^l \mathcal{W}_\nu^l(t). \tag{10.21}$$

Substituting (10.21) in (10.8) and using Hadamard product notation, we obtain

$$N_\nu^l(\mathcal{W}_\nu^l(t)) = \frac{1}{2}(\Psi_\nu^l)^\top K(\Psi_\nu^l \mathcal{W}_\nu^l(t)) \circ (\Psi_\nu^l \mathcal{W}_\nu^l(t)),$$

and hence the reduced-order model is described by

$$\dot{\mathcal{W}}_\nu^l(t) + \nu S_\nu^l \mathcal{W}_\nu^l(t) + \frac{1}{2}(\Psi_\nu^l)^\top K(\Psi_\nu^l \mathcal{W}_\nu^l(t)) \circ (\Psi_\nu^l \mathcal{W}_\nu^l(t)) - \nu L_\nu^l U(t) = F_\nu^l(t), \tag{10.22}$$

which concludes the proof.

10.3 Error Estimates and Sliding Mode Control Design using Reduced-order Model

The use of a high-order controller for real-time control of a complex system is not practical due to the computational complexity involved in both control design process and its implementation. Therefore, the need for a low-order controller is inevitable. The "reduce then design" approach employs the reduced-order state space model given by (10.17) with an adequately small value of l for control design purposes while considering model variations. In this section, we explain in detail how a sliding mode controller is designed on the basis of the POD-based reduced-order model determined at the viscosity ν_0 to guarantee stability of the closed-loop interconnection of the sliding mode controller and the original full-order model for varying viscosity ν.

10.3.1 Preliminaries for Sliding Mode Control Design

To design a sliding mode controller, a sliding surface should be defined first. To this end, we refer to the state-space representation of the system in (10.4) accompanied with the full-order system model outputs (or the measurement equations) as

$$\begin{aligned} \dot{\mathcal{W}}_\nu(t) &= A\mathcal{W}_\nu(t) + \mathbf{h}_\nu(t, \mathcal{W}_\nu(t), U(t)) \\ Y_\nu(t) &= C\mathcal{W}_\nu(t), \end{aligned} \tag{10.23}$$

where $\mathcal{W}_\nu(t)$ represents the state vector of the full-order model, and C represents the system measurement matrix. For a nominal value of the system parameter, i.e., the viscosity ν_0, the set of basis functions $\Psi^l_{\nu_0}$ is used to obtain the nominal reduced-order model in the state-space form as

$$\begin{aligned} \dot{\mathcal{W}}^l_{\nu_0}(t) &= A^l_{\nu_0}\mathcal{W}^l_{\nu_0}(t) + \mathbf{g}_{\nu_0}(t, \mathcal{W}^l_{\nu_0}(t), U(t)), \\ Y_{\nu_0}(t) &= C^l_{\nu_0}\mathcal{W}^l_{\nu_0}(t), \end{aligned} \tag{10.24}$$

where $\mathcal{W}^l_{\nu_0}(t)$ represents the state vector of the nominal reduced-order model. The reduced measurement matrix $C^l_{\nu_0}$ is obtained as

$$C^l_{\nu_0} = C\Psi^l_{\nu_0}. \tag{10.25}$$

The control objective is to track a reference trajectory via a reduced-order model based sliding surface designed in a priori known nominal value of the system parameter ν_0. The idea here is to design a reduced-order model based control law associated with the nominal value ν_0 that ensures the desired reference tracking for the full order model while the parameter ν changes over time. A sliding surface is introduced to ensure the output tracking by defining the surface via dynamics of the full-order model where the control input is imposed via the dynamics of the reduced order model. The sliding surface on the dynamics of the full-order model is defined as

$$S(t) = Y_\nu(t) - r(t) = C\mathcal{W}_\nu(t) - r(t), \tag{10.26}$$

where $S(t) = [S_1(t) \quad S_2(t)]^\top$, $Y_\nu(t) = [y_{\nu,1}(t) \quad y_{\nu,2}(t)]^\top$ is the system output, and $r(t) = [r_1(t) \;\; r_2(t)]^\top$ is the reference signal. The sliding surface is described by the reduced-order model as

$$S^l(t) = C\Psi^l_{\nu_0}\mathcal{W}^l{}_{\nu_0,\nu}(t) + C\delta^\nu_{\nu_0}\mathcal{W} - r(t), \tag{10.27}$$

where $\delta^\nu_{\nu_0}\mathcal{W} = \mathcal{W}_\nu(t) - \Psi^l_{\nu_0}\mathcal{W}^l{}_{\nu_0,\nu}(t)$. The sliding mode controller (SMC) needs to have real-time access to the reduced-order model states. Thus, a nonlinear low-order functional observer is designed using the method proposed in [5] to estimate the states of the reduced-order model. The design of the observer will be described at the end of this section. SMC law usually includes a switching control law and an equivalent control law [127]. The switching control law that is designed with respect to the definition of the reduced-order surface drives the system states towards a predefined sliding surface in the presence of model uncertainties. It will be shown that the switching control law that implements the dynamics on the reduced-order surface effectively stabilizes the main system. Furthermore, the equivalent control law guarantees that the system states keep sliding on the sliding surface and converge to the surface. The general control law is considered as

$$U(t) = u_{eq}(t) + u_{sw}(t), \tag{10.28}$$

where $u_{eq}(t)$ and $u_{sw}(t)$ represent the equivalent control and the switching control laws, respectively. To construct the equivalent dynamics, the full-order model is considered while states are

sliding on the defined surface. In fact, the equivalent control provides an action only when the system states are on the surface. Taking the derivative of (10.26) results in

$$\dot{S}(t) = \dot{Y}_\nu(t) - \dot{r}(t) = C\dot{\mathcal{W}}_\nu(t) - \dot{r}(t). \tag{10.29}$$

Since we do not have online access to the states of the full-order model for varying viscosity ν, the dynamics of the sliding surface is defined in terms of the states of the reduced-order model as

$$\begin{aligned}\dot{S}^l = C^l\dot{\mathcal{W}}^l{}_{\nu_0,\nu}(t) + \big(C\dot{\mathcal{W}}_\nu(t) - C^l\dot{\mathcal{W}}^l{}_{\nu_0,\nu}(t)\big) - \dot{r}(t) = CA\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t) + \Delta^\nu_{\nu_0}\mathcal{W} - \\ \frac{1}{2}CM^{-1}K(\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t)) \circ (\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t)) + \Delta^\nu_{\nu_0}\mathbf{f} + CM^{-1}F(t) + \nu CM^{-1}LU(t) - \dot{r}(t),\end{aligned} \tag{10.30}$$

where the terms representing uncertainties are represented as

$$\Delta^\nu_{\nu_0}\mathcal{W} = CA\bigg(\mathcal{W}_\nu(t) - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t)\bigg), \tag{10.31}$$

and

$$\Delta^\nu_{\nu_0}\mathbf{f} = \mathbf{f}(\mathcal{W}_\nu(t)) - \mathbf{f}(\Psi^l_{\nu_0}\mathcal{W}_{\nu_0,\nu}(t)) = -\frac{1}{2}CM^{-1}K\bigg((\mathcal{W}_\nu(t)\circ\mathcal{W}_\nu(t)) - (\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t))\circ(\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t))\bigg). \tag{10.32}$$

The nonlinear term described by (10.32) is assumed to be locally Lipschitz with respect to $\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}$ and $\mathcal{W}_\nu$ in a region $\mathcal{D}$, i.e., for any $\mathcal{W}_\nu(t), \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t) \in \mathcal{D}$ [5]

$$\big\|\mathbf{f}(\mathcal{W}_\nu) - \mathbf{f}(\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu})\big\| \leq \gamma_h\big\|\mathcal{W}_\nu - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\big\|,$$

where $\|\ .\ \|$ represents the 2-norm and γ_h is the non-negative Lipschitz constant. Hence, two model uncertainty terms are represented in terms of the states of the full-order and reduced-order models.

10.3.2 Error Estimates for a Variation of the Viscosity

The uncertainty terms are inherently resulted from the sensitivity of the POD method with respect to the variation in the system parameter ν. The main challenge is to represent the confidence region as a function of the changing parameter. The following analysis is carried out to present a priori

estimate for the squared error in the Hilbert space for the reference set of POD basis functions $\Psi^l_{\nu_0}$. The error caused by any variation in the parameter value ν can be written as

$$\left\|\mathcal{W}_\nu - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\right\| = \left\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0} + \mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0} + \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\right\|. \tag{10.33}$$

After applying CauchySchwarz inequality multiple times, we obtain

$$\left\|\mathcal{W}_\nu - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\right\|^2 \leq 2\left\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\right\|^2 + 4\left\|\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2 + 4\left\|\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\right\|^2. \tag{10.34}$$

As seen from the right hand side of the inequality above, the boundedness of the each error term needs to be investigated to essentially evaluate the bound on $\left\|\mathcal{W}_\nu - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\right\|$.

Lemma 10.3.1 *The estimation error representing the variation in the states of the full-order model associated with changing ν in a given time interval $(0, T)$ is bounded by*

$$\left\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\right\|^2_M \leq \Omega|\nu - \nu_0|^2, \tag{10.35}$$

where

$$\Omega = \frac{\left(\left\|M^{-1}S\right\|\frac{1}{4\eta}k^2 + \frac{1}{4\eta}\gamma_2\left\|M^{-1}L\right\|\right)T}{\gamma_1 + \eta\gamma_2\left\|M^{-1}L\right\|}\exp\left(\left(2\gamma_1 + 2\eta\gamma_2\left\|M^{-1}L\right\|\right)T - 1\right), \tag{10.36}$$

where γ_1 and γ_2 are non-negative Lipschitz constants and $k = \left\|\mathcal{W}_{\nu_0}\right\|_{L_\infty}$.

Proof 8 *We start by subtracting the full-order models associated with the nominal and new parameters ν and ν_0 as*

$$\dot{\mathcal{W}}_\nu - \dot{\mathcal{W}}_{\nu_0} = -\nu M^{-1}S\mathcal{W}_\nu + \nu_0 M^{-1}S\mathcal{W}_{\nu_0} + \boldsymbol{f}(\mathcal{W}_\nu) - \boldsymbol{f}(\mathcal{W}_{\nu_0}) + (\nu - \nu_0)M^{-1}LU(t). \tag{10.37}$$

By multiplying both sides by $\mathcal{W}_\nu - \mathcal{W}_{\nu_0}$, the following is obtained

$$\left\langle \dot{\mathcal{W}}_\nu - \dot{\mathcal{W}}_{\nu_0}, \mathcal{W}_\nu - \mathcal{W}_{\nu_0}\right\rangle_M = \left\langle -M^{-1}S\Big(\nu(\mathcal{W}_\nu - \mathcal{W}_{\nu_0}) + (\nu - \nu_0)\mathcal{W}_{\nu_0}\Big) + \boldsymbol{f}(\mathcal{W}_\nu) - \boldsymbol{f}(\mathcal{W}_{\nu_0}) + (\nu - \nu_0)M^{-1}LU(t), \mathcal{W}_\nu - \mathcal{W}_{\nu_0}\right\rangle_M. \tag{10.38}$$

This can be rewritten in the following form using M induced norm

$$\frac{1}{2}\frac{d}{dt}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 \leq \big\|M^{-1}S\big\|\Big(-\nu\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + |\nu-\nu_0|\big\|\mathcal{W}_{\nu_0}\big\|_{L_\infty}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M\Big) + \\ \big\|\boldsymbol{f}(\mathcal{W}_\nu) - \boldsymbol{f}(\mathcal{W}_{\nu_0})\big\|\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M + \big\|M^{-1}L\big\|\big\|(\nu-\nu_0)U(t)\big\|\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M \quad (10.39)$$

where the matrix norm induced by the vector norm $\|\cdot\|_M$ is defined as $\|B\| = \max\|Bu\|_M$, $\|u\|_M = 1$. For any admissible control input $U(t)$ and a locally Lipschitz nonlinearity $\boldsymbol{f}$ in $\mathcal{W}_\nu, \mathcal{W}_{\nu_0} \in \mathcal{D}$, (10.39) results in

$$\frac{1}{2}\frac{d}{dt}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 \leq \big\|M^{-1}S\big\|\Big(-\nu\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + |\nu-\nu_0|\big\|\mathcal{W}_{\nu_0}\big\|_{L_\infty}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M\Big) + \\ \gamma_1\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + \gamma_2\big\|M^{-1}L\big\||\nu-\nu_0|\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M, \quad (10.40)$$

Applying Youngs inequality results in

$$\frac{1}{2}\frac{d}{dt}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 \leq \big\|M^{-1}S\big\|\Big(-\nu\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + \frac{\zeta}{2}|\nu-\nu_0|^2\big\|\mathcal{W}_{\nu_0}\big\|_{L_\infty}^2 + \frac{1}{2\zeta}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2\Big) + \\ \gamma_1\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + \frac{1}{2}\gamma_2\big\|M^{-1}L\big\|\Big(\zeta|\nu-\nu_0|^2 + \frac{1}{\zeta}\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2\Big). \quad (10.41)$$

By choosing $\zeta = \frac{1}{2\eta}$, where $\eta \in \mathbb{R}$ is close to zero, it is ensured that $-\nu + \frac{1}{2\zeta} \leq 0$. Hence, we obtain

$$\frac{d}{dt}\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 \leq \Big(2\gamma_1 + 2\eta\gamma_2\big\|M^{-1}L\big\|\Big)\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 + \Big(\big\|M^{-1}S\big\|\frac{1}{2\eta}k^2 + \\ \frac{1}{2\eta}\gamma_2\big\|M^{-1}L\big\|\Big)|\nu-\nu_0|^2. \quad (10.42)$$

Using the Gronwall lemma over an interval $(0,t) \subset (0,T)$, we have

$$\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2 \leq \frac{\Big(\big\|M^{-1}S\big\|\frac{1}{4\eta}k^2 + \frac{1}{4\eta}\gamma_2\big\|M^{-1}L\big\|\Big)|\nu-\nu_0|^2 T}{\gamma_1 + \eta\gamma_2\big\|M^{-1}L\big\|}\exp\Big(\big(2\gamma_1 + 2\eta\gamma_2\big\|M^{-1}L\big\|\big)T - 1\Big). \quad (10.43)$$

and this concludes the proof for the boundedness of $\big\|\mathcal{W}_\nu - \mathcal{W}_{\nu_0}\big\|_M^2$.

Lemma 10.3.2 *The POD-Galerkin error $\big\|\mathcal{W}_{\nu_0} - \Psi_{\nu_0}^l\mathcal{W}_{\nu_0}^l\big\|$ is bounded by*

$$\big\|\mathcal{W}_{\nu_0} - \Psi_{\nu_0}^l\mathcal{W}_{\nu_0}^l\big\|_M^2 \leq \big(1 + \xi\nu_b^2\big\|\Psi_{\nu_0}^l(\Psi_{\nu_0}^l)^\top S\big\|^2 + \gamma_1^2\big\|\Psi_{\nu_0}^l(\Psi_{\nu_0}^l)^\top\big\|^2\big)T\sum_{i=l+1}^{N}\lambda_{\nu_0,i} + \\ \big\|M^{-1} - \Psi_{\nu_0}^l(\Psi_{\nu_0}^l)^\top\big\|G^2. \quad (10.44)$$

Proof 9 *The POD-Galerkin error is broken into the Galerkin projection error* $\left\|\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}\right\|$ *and* $\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|$*. The first error term was proven in [150] to be bounded on the time interval* $(0, T)$ *as*

$$\left\|\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}\right\| \leq T \sum_{i=l+1}^{N} \lambda_{\nu,i}. \tag{10.45}$$

To investigate the boundedness of the POD-Galerkin error $\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\|$*, the projection operator* $\mathcal{P}^l_{\psi_{\nu_0}}$ *is written in the matrix form as* $\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top M$ *where* $\Psi^l_{\nu_0} = \{\psi^l_{\nu_0,1}, \psi^l_{\nu_0,2}, \ldots, \psi^l_{\nu_0,l}\}$*. From the models (10.4) and (10.17), we obtain*

$$\begin{aligned}
&\langle \frac{d}{dt}(\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}^l_{\nu_0}) + \frac{d}{dt}(\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}), \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\rangle_M = \\
&\Big\langle \Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top A_{\nu_0}(\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}) + (M^{-1}A_{\nu_0} - \Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top A_{\nu_0})\mathcal{W}_{\nu_0} + (M^{-1} - \Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top)\boldsymbol{h}_\nu(t, \mathcal{W}_{\nu_0}(t), U(t)) + \\
&\qquad \Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top \Big(\boldsymbol{h}_{\nu_0}(t, \mathcal{W}_{\nu_0}(t), U(t)) - \boldsymbol{h}_{\nu_0}(t, \Psi^l_{\nu_0}\mathcal{W}_{\nu_0}(t), U(t))\Big), \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\Big\rangle_M.
\end{aligned} \tag{10.46}$$

Using the commutativity of the time derivative and the projection operator $\mathcal{P}^l_{\psi_{\nu_0}}$ *remote space, it is concluded that*

$$\langle \frac{d}{dt}(\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}^l_{\nu_0}), \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\rangle_M = 0. \tag{10.47}$$

For an admissible and bounded control input $U(t)$*, the full-order model (10.23) is bounded by* $G = \left\|A_{\nu_0}\mathcal{W}_{\nu_0}(t) + \boldsymbol{h}_{\nu_0}(t, \mathcal{W}_{\nu_0}(t), U(t))\right\|_\infty$ *[150]. Next, using the properties of the weighted inner product and applying the Young's inequality considering locally Lipschitz nonlinearity* $\boldsymbol{h}_{\nu_0}$*, we obtain*

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M \leq \left(\frac{\xi}{2}\nu_0^2\left\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\right\|^2\left\|\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}\right\|^2_M + \frac{1}{2\xi}\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M - \right. \\
&\left.\nu_0\left\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\right\|\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M\right) + \left(\frac{1}{2}\left\|M^{-1} - \Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\right\|^2 G^2 + \frac{1}{2}\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M\right) + \\
&\left(\frac{1}{2}\gamma_1^2\left\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\right\|^2\left\|\mathcal{W}_{\nu_0} - \mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}^l_{\nu_0}\right\|^2_M + \frac{1}{2}\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M + \gamma_1\left\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\right\|\left\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0} - \Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\right\|^2_M\right).
\end{aligned} \tag{10.48}$$

After rearranging (10.48), we obtain

$$\frac{d}{dt}\big\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\big\|^2_M \leq \bigg(\xi\nu_0^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|^2+ \gamma_1^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|^2\bigg)\big\|\mathcal{W}_{\nu_0}-\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}\big\|^2_M + \big\|M^{-1}-\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|G^2+ \bigg(-2\nu_0\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|+\frac{1}{\xi}+2+2\gamma_1\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|\bigg)\big\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\big\|^2_M. \quad (10.49)$$

Using (10.45) and the Gronwall lemma, the following inequality is obtained

$$\big\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\big\|^2_M \leq \bigg(\big(\xi\nu_0^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|^2+\gamma_1^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|^2\big)T\sum_{i=l+1}^{N}\lambda_{\nu,i}+ \big\|M^{-1}-\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|G^2\bigg)\exp\bigg(\big(-2\nu_0\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|+\frac{1}{\xi}+2+2\gamma_1\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|\big)t\bigg). \quad (10.50)$$

To ensure the negativity of the term inside the exponential function, the following inequality should hold

$$\nu_0 > \frac{\frac{1}{\xi}+2+2\gamma_1\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|}{\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|}. \quad (10.51)$$

If inequality (10.51) holds true for $\nu_0=\nu_b$, we can obtain the bound on the error as follows

$$\big\|\mathcal{P}^l_{\psi_{\nu_0}}\mathcal{W}_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\big\|^2_M \leq \big(\xi\nu_b^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|^2+\gamma_1^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|^2\big)T\sum_{i=l+1}^{N}\lambda_{\nu_0,i}+\big\|M^{-1}-\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|G^2. \quad (10.52)$$

By combining (10.53) and (10.45), we obtain

$$\big\|\mathcal{W}_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}\big\|^2_M \leq \big(1+\xi\nu_b^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|^2+\gamma_1^2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|^2\big)T\sum_{i=l+1}^{N}\lambda_{\nu_0,i}+\big\|M^{-1}-\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top\big\|G^2. \quad (10.53)$$

This concludes the proof for the boundedness of the POD-Galerkin error.

Remark 10.1 *The bound on the last term of the error in (10.34), $\big\|\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\big\|^2$, is obtained as*

$$\big\|\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0}-\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}\big\|^2_M \leq \bar{\Omega}|\nu-\nu_0|^2, \quad (10.54)$$

where

$$\bar{\Omega}=\bigg(\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top S\big\|\frac{1}{4\nu_0}\bar{k}^2+\frac{1}{4\nu_0}\gamma_2\big\|\Psi^l_{\nu_0}(\Psi^l_{\nu_0})^\top L\big\|\bigg)|\nu-\nu_0|^2T. \quad (10.55)$$

The proof follows the same procedure as presented in Lemma 10.3.1.

The following theorem ensures the boundedness of the model uncertainties $A\Delta^{\nu}_{\nu_0}\mathcal{W}$ and $\Delta^{\nu}_{\nu_0}\mathbf{g}$.

Theorem 10.3.3 *The error caused by the estimation of the full-order state $\mathcal{W}$ through a priori chosen sequence of the POD eigenvalues $\{\lambda_i\}_{i=1}^{l}$ and basis functions $\Psi^{l}_{\nu_0}$ at the nominal viscosity parameter ν_0 can be estimated and bounded with respect to the variation of ν as*

$$\big\|\mathcal{W}_{\nu}(t) - \Psi^{l}_{\nu_0}\mathcal{W}^{l}_{\nu_0,\nu}(t)\big\| \leq 2\big(1+\xi\nu_b^2\big\|\Psi^{l}_{\nu_0}(\Psi^{l}_{\nu_0})^{\top}S\big\|^2 + \gamma_1^2\big\|\Psi^{l}_{\nu_0}(\Psi^{l}_{\nu_0})^{\top}\big\|^2\big)T\sum_{i=l+1}^{N}\lambda_{\nu_0,i} + \big\|M^{-1} - \Psi^{l}_{\nu_0}(\Psi^{l}_{\nu_0})^{\top}\big\|G^2 + (2\Omega + 4\bar{\Omega})|\nu-\nu_0|^2. \quad (10.56)$$

Proof 10 *From (10.34), Lemma 10.3.1 and Lemma 10.3.2, we obtain the inequality (10.56).*

Inequality (10.56) gives an estimate of the error associated with the model reduction and varying parameter described by the POD basis functions at the nominal viscosity ν_0. As seen from (10.56), the first two terms will vanish for an N^{th} order reduced model. In fact, the first two terms represent the error associated with the reduced-order model and the last term takes into account the effect of the changing parameter ν.

Remark 10.2 *The model uncertainties $\Delta^{\nu}_{\nu_0}\boldsymbol{f}$ and $\Delta^{\nu}_{\nu_0}\mathcal{W}$ associated with the estimation of the POD bases Ψ_{ν_0} at the nominal viscosity ν_0 are bounded by variation of the system parameter ν and a priori known set of system matrices as described in Theorem 10.3.3, $\big\|\Delta^{\nu}_{\nu_0}\mathcal{W}\big\| < \mathcal{B}_1$ and $\big\|\Delta^{\nu}_{\nu_0}\boldsymbol{f}\big\| < \mathcal{B}_2$. The error bound is calculated offline with respect to the reduced and full-order models at the nominal value ν_0.*

10.3.3 Stability Analysis of the Closed-loop System with the Proposed Sliding Model Controller

An SMC law is synthesized here to drive the system trajectories onto the predefined sliding surface (10.26) in a finite time while taking into account the model uncertainties due to the changing parameter ν. The main objective can be stated as designing a control law using the reduced-order model at the nominal viscosity ν_0 with the bounded model uncertainties as obtained in the previous

section. The following theorem gives an SMC law to guarantee the stability of the full-order model.

Theorem 10.3.4 *The asymptotic stability of the system described by (10.23) is guaranteed for the sliding surface (10.26) with the following sliding mode control law*

$$u(t) = (CM^{-1}L)^{-1}\bigg(\dot{r}(t) - C\bigg(A\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t) - \frac{1}{2}M^{-1}K(\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t)) \circ (\Psi^l_{\nu_0}\mathcal{W}^l_{\nu_0,\nu}(t)) + M^{-1}F(t)\bigg)\bigg) - (CM^{-1}L)^{-1}(\lambda_1 S^l + \lambda_2(\mathcal{B}_1 + \mathcal{B}_2)\,\mathrm{sign}(S^l)), \quad (10.57)$$

where $\lambda_1 > 0$ and $\lambda_2 > 1$ are the sliding mode parameters and $\mathrm{sign}(\cdot)$ is the sign function.

Proof 11 *In order to investigate the asymptotic stability of the closed-loop system with the sliding mode control law in (10.57), following Lyapunov function is considered*

$$V(t) = \frac{1}{2}{S^l}^\top S^l, \quad (10.58)$$

where S^l is the sliding surface defined in (10.27). For the chosen Lyapunov function, we should have

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} < 0 \quad \textit{or} \quad {S^l}^\top \dot{S}^l < 0, \quad (10.59)$$

in a neighborhood of the surface given by $S^l = 0$. Substituting (10.27) and (10.30) into (10.59), we obtain

$${S^l}^\top \dot{S}^l = {S^l}^\top \big(C\Psi^l_{\nu_0}\dot{\mathcal{W}}^l_{\nu_0,\nu}(t) + C\delta^\nu_{\nu_0}\dot{\mathcal{W}} - \dot{r}(t)\big) = {S^l}^\top \bigg(- \lambda_1 S^l + (\Delta^\nu_{\nu_0}\boldsymbol{f} + \Delta^\nu_{\nu_0}\mathcal{W}) - \lambda_2(\mathcal{B}_1 + \mathcal{B}_2)\,\mathrm{sign}(S^l)\bigg). \quad (10.60)$$

According to Remark 10.2, $\left\|\Delta^\nu_{\nu_0}\boldsymbol{f} + \Delta^\nu_{\nu_0}\mathcal{W}\right\| < (\mathcal{B}_1 + \mathcal{B}_2)$, and hence, we have

$${S^l}^\top \bigg(\Delta^\nu_{\nu_0}\boldsymbol{f} + \Delta^\nu_{\nu_0}\mathcal{W} - \lambda_2(\mathcal{B}_1 + \mathcal{B}_2)\,\mathrm{sign}(S^l)\bigg) < 0, \quad (10.61)$$

where λ_2 is a diagonal matrix with $\lambda_{11} > 1$ and $\lambda_{22} > 1$. This concludes the negativity of the time derivative of Lyapunov function $\dot{V} < 0$.

As observed from (10.60), the defined surface is a function of the states of the reduced-order model. Therefore, the reduced-order controller is capable of stabilizing the full-order model in the presence of the model uncertainties arising from both model reduction and changing parameter ν. The constants in the proposed SMC law are chosen by trial and error considering the trade-off between the reaching time and chattering. These constants can be considered large enough when the trajectory is far from the switching surface (so that the reaching time is short), and then as small as desired in order to limit the chattering.

10.3.4 FUNCTIONAL OBSERVER DESIGN

The linear functionals of the full-order model states can be estimated by a functional observer. The states of the reduced-order model can be written as linear functionals of the full-order model states (FE model states), and hence, a functional observer can be designed to estimate the states of the reduced-order model leading to a reduced computational cost.

The reduced-order model described by (10.17) is obtained from the discretized Burgers' equation with a locally Lipschitz nonlinearity with respect to $\mathcal{W}_\nu^l$ in a region $\mathcal{D}$, i.e., for any $\mathcal{W}_{\nu,1}^l(t), \mathcal{W}_{\nu,2}^l(t) \in \mathcal{D}$ [5]

$$\left\|\mathbf{g}(\mathcal{W}_{\nu,1}^l, U^*) - \mathbf{g}(\mathcal{W}_{\nu,2}^l, U^*)\right\| \leq \gamma_d \left\|\mathcal{W}_{\nu,1}^l - \mathcal{W}_{\nu,2}^l\right\|,$$

where $\| \, . \, \|$ represents the 2-norm, U^* is an admissible control sequence and γ_d is the nonnegative Lipschitz constant. The proposed observer takes the following form

$$\dot{\hat{\mathcal{W}}}_\nu^l(t) = A^l \hat{\mathcal{W}}_\nu^l(t) + \mathbf{g}(\hat{\mathcal{W}}_\nu^l(t), U(t)) + L_o(Y(t) - C^l \hat{\mathcal{W}}_\nu^l(t)), \tag{10.62}$$

where $\hat{\mathcal{W}}_\nu^l$ is the estimate of $\mathcal{W}_\nu^l$ and L_o is the observer matrix gain chosen such that the observer error system defined by (10.63)-(10.64) is asymptotically stable

$$e^l(t) = \mathcal{W}_\nu^l(t) - \hat{\mathcal{W}}_\nu^l(t), \tag{10.63}$$

and

$$\dot{e}^l(t) = (A^l - L_o C^l) e^l(t) + \mathbf{g}(\mathcal{W}_\nu^l(t), U(t)) - \mathbf{g}(\hat{\mathcal{W}}_\nu^l(t), U(t)). \tag{10.64}$$

As shown in [5], a linear matrix inequality (LMI)-based sufficient condition can be determined to maximize γ_d while a stabilizing L_o is obtained. The observer of order l in (10.62) is used to estimate $\mathcal{W}_\nu^l$ considering that the quadratic nonlinearities in the reduced-order model described by (10.17) are locally Lipschitz.

10.3.5 Summary of the Proposed SMC-based Control Design Approach

The implementation of the developed model reduction approach and sliding model controller is summarized below:

1. Find the equivalent lumped-parameter model of the Burgers' PDE as in (10.9) by using FE method.

2. Develop the reduced-order model by utilizing the continuous POD approach, and then describe it in the state-space form (10.17) in parameter value ν_0.

3. Design a functional observer as in (10.62) to estimate the states of the reduced-order model for ν_0.

4. Estimate the bounds on the model errors as given in (10.56) and Remark 10.2 for varying parameter ν.

5. Design the sliding mode controller based on the reduced-order model by

   (a) defining the sliding surface for the reduced-order model as in (10.27) and finding equivalent control law u_{eq}, and

   (b) finding the control law $u_{sw} + u_{eq}$ as in (10.57) to guarantee the closed-loop system stability.

10.4 Simulation Results and Discussion

In this section, we illustrate some of the simulation results of our numerical studies and further provide a discussion on the accuracy of the derived reduced-order models, as well as the performance of the closed-loop system achieved from the implementation of the designed sliding mode controller.

10.4.1 Open-loop System Simulation Results to Evaluate the Reduced-oder Model Accuracy

In order to assess the accuracy of the presented model reduction method, an example of a viscous Burgers' equation is considered here. The forcing term in (10.1a) is considered to be

$$f(t,x) = -\exp(-t)\sin(\pi x). \tag{10.65}$$

The initial condition is assumed to be

$$w_0(x) = \begin{cases} 0.45 - 0.5\cos(8\pi x) - 0.05\cos(16\pi x), & \text{for } x \in (0, \frac{1}{4}] \\ 0, & \text{otherwise.} \end{cases}$$

Also, a set of sinusoidal boundary conditions covering frequencies up to 75Hz are used to simulate a rich snapshot matrix that is later used to obtain the POD basis vectors. Running this kind of boundary regime excites a reasonably large number of dynamical constituents of the Burgers system [104]. Also, for the simulation purpose N is chosen as 160. As the first step, the eigenvalues corresponding to the POD method are extracted for the nominal value of $\nu_0 = 0.01$ and shown in a descending order in Figure 10.1. As observed, the first seven eigenvalues capture more than 99% of the system's total energy. Also, the percentage of the total energy captured by the chosen number of eigenvalues introduced in (10.15) is shown in Figure 10.1. The numerical simulation of the open-loop system by both FEM and 7^{th} POD approach are illustrated for $\nu = 0.01$. As seen in Figure 10.2, the POD approach represents a close match with the solution by finite element method. In order to observe the difference between the behavior of the models with varying viscosity, Figure

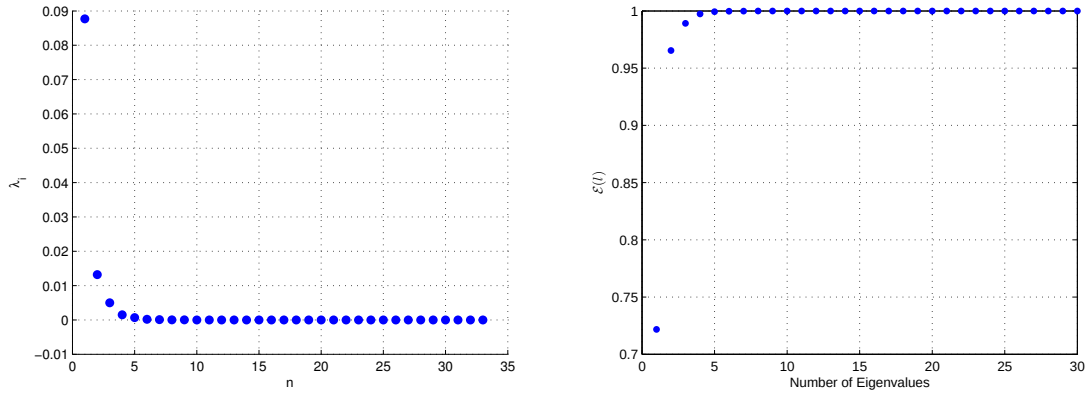


Figure 10.1: The extracted eigenvalues corresponding to the POD eigenvectors (left); The percentage of the energy captured by different numbers of chosen eigenvalues (right).

10.3 shows the open-loop simulation result of the Burgers' equation for $\nu = 0.1$ and the simulated response with the extracted eigenvectors at $\nu_0 = 0.01$. The simulated response using POD shows a different profile of the flow due to the discrepancy between true eigenvectors at ν and extracted basis at ν_0.

10.4.2 Closed-loop Simulation Results Using the Proposed Sliding Mode Controller

The discretized full-order model obtained from FEM is utilized to validate the designed robust sliding mode controller (SMC) consisting of equivalent and switching control laws in order to examine the performance of the proposed SMC law in tracking a given reference trajectory in the presence of the model uncertainties. From the fluid mechanics point of view, this can be seen as the problem of controlling the flux on the boundaries to reach the desired flow velocity at the desired points while the model uncertainties exist due to the varying viscosity and uncertainty in the reduced-order model. It is assumed that the viscosity changes from the nominal value $\nu_0 = 0.01$ to $\nu = 0.1$. The Burgers' equation with the same initial condition as given in the previous section

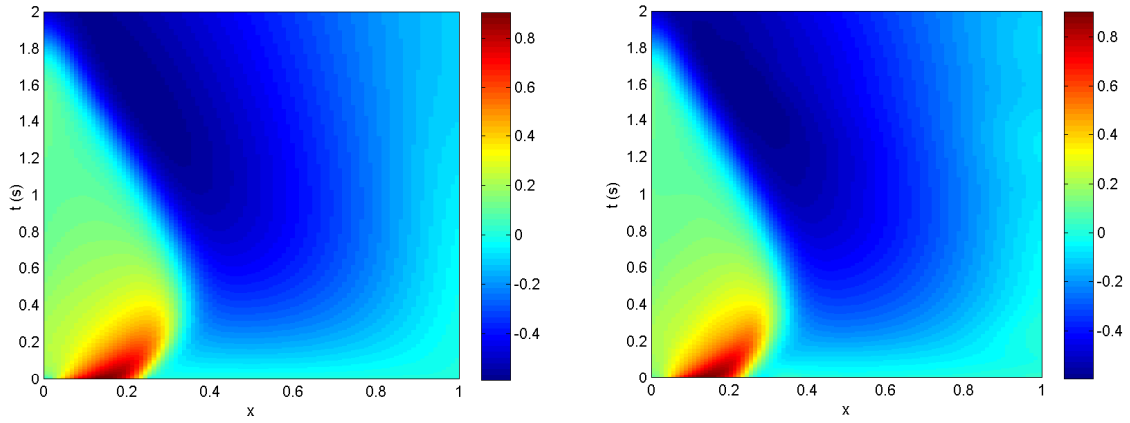


Figure 10.2: Finite element solution for 160 spatial points (left), and POD solution with 7 basis functions (right) for $\nu = 0.01$.

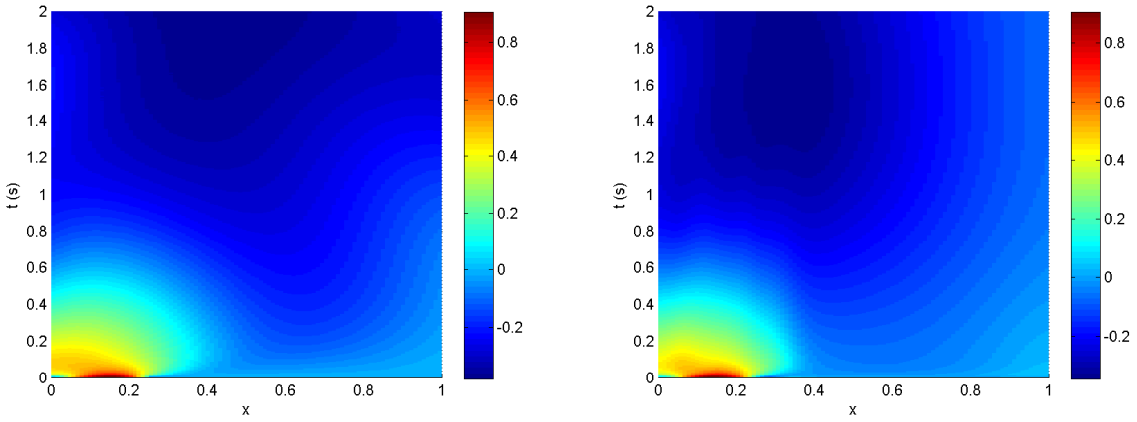
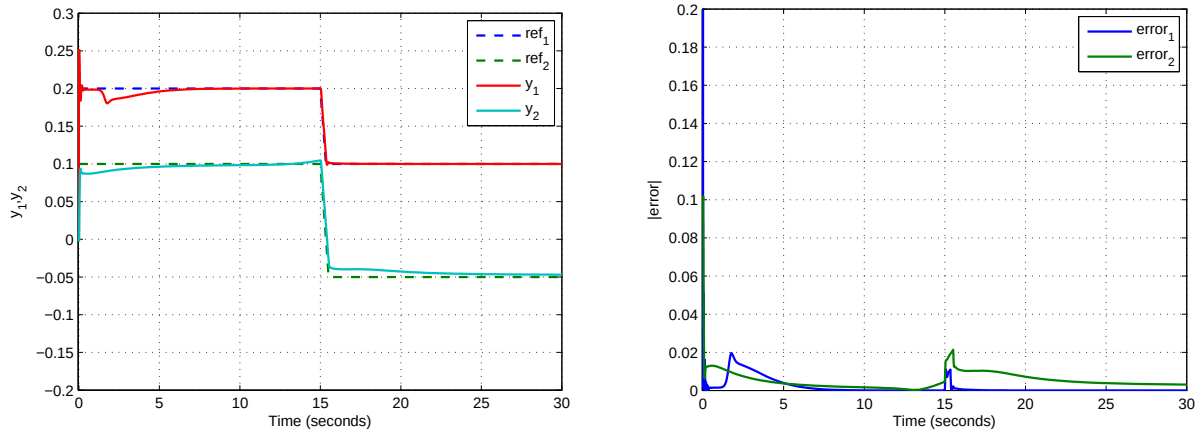


Figure 10.3: Finite element solution for 160 spatial points (left), and POD solution with 7 basis functions extracted at $\nu_0 = 0.01$ (right) for $\nu = 0.1$.



(a) System outputs

(b) Absolute error signals

Figure 10.4: (a) System outputs and reference inputs for a ramp reference function; (b) The absolute error signals associated with the system outputs.

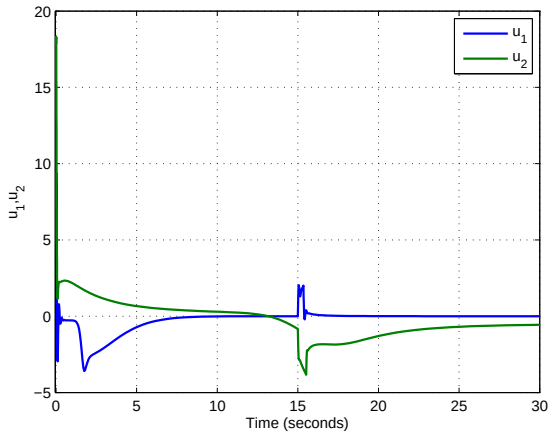


Figure 10.5: The control inputs generated by the proposed SMC law to track a ramp function.

is used to validate the proposed SMC design approach. The two components of the SMC laws are obtained from the 7^{th} order reduced model. Control outputs in flow control problems are typically close to the boundaries, hence, the measurement devices are considered to be placed on the

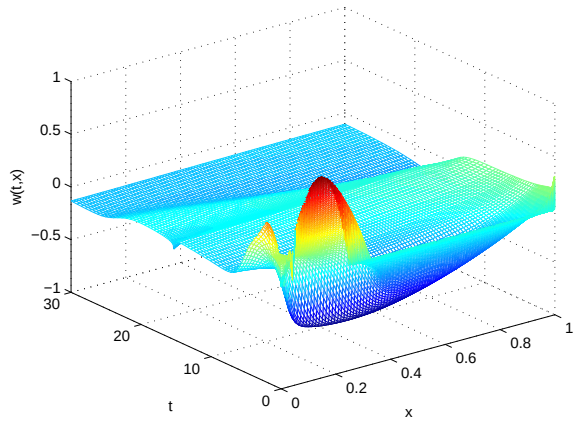
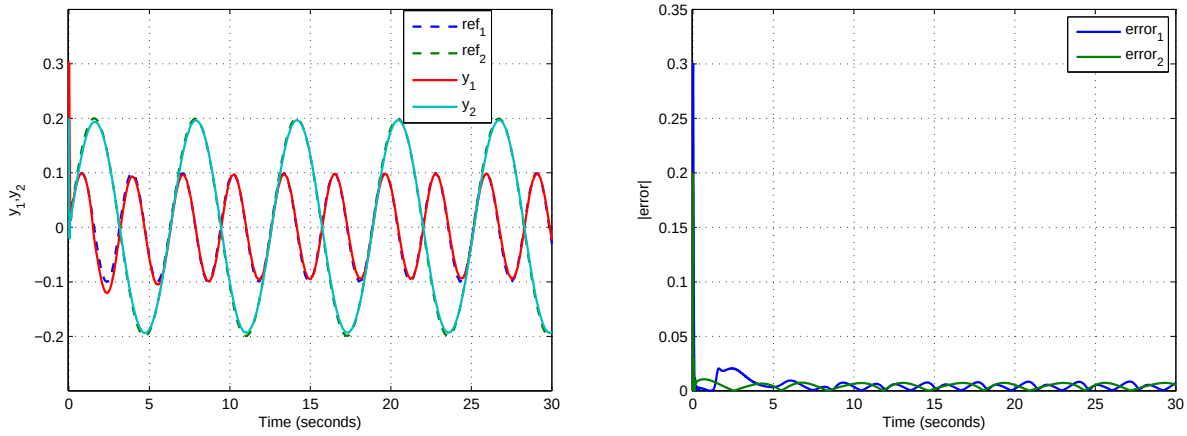


Figure 10.6: Velocity response of the full-order model with the proposed SMC law to track a ramp function.



(a) System outputs

(b) Absolute error signals

Figure 10.7: (a) System outputs and the reference inputs for the given sinusoidal function; (b) The absolute error signals associated with the system outputs.

points near boundaries at $y_1 = 0.895$ and $y_2 = 0.074$ to collect the flow velocity measurement. Two different functions are considered as reference inputs, a ramp and a sinusoidal function. Fig-

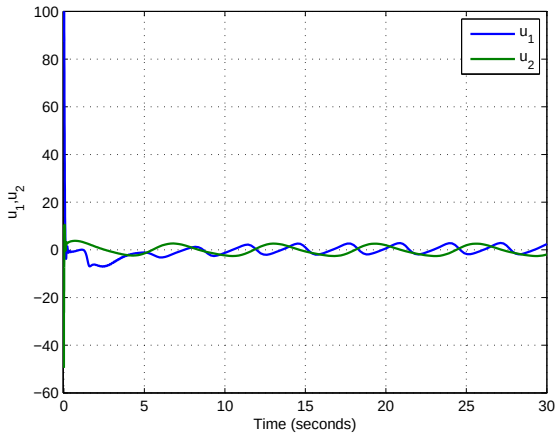


Figure 10.8: The control inputs generated by the proposed SMC law to track a sinusoidal function.

ures 10.4 and 10.5 illustrate the tracking performance and the control inputs for a ramp function, respectively. As described earlier, the constant parameters of the switching control law in (10.57) are chosen by trial and error in a way that a reasonable trade-off between chattering and reaching time can be achieved. After trying different combinations of the switching control parameters, the best results are obtained using the parameters $\lambda_{11} = 4.3$ and $\lambda_{22} = 9.3$ given the ramp reference signal. The simulation result of the full-order model implementing the SMC is shown in Figure 10.6. The tracking performance and control inputs for a given sinusoidal reference signal are shown in figures 10.7 and 10.8, respectively. The corresponding sliding mode controller parameters are tuned as $\lambda_1 = 9.6$ and $\lambda_2 = 19.8$. The simulation result of the full-order model is shown in Figure 10.9. As observed from figures 10.4 and 10.7, the proposed SMC law illustrates a very good tracking performance in the presence of model uncertainties. Uncertainties in the problem in hand are primarily due to the discrepancy between the full-order and reduced-order models, where the eigenfunctions obtained at ν_0 to find the reduced-order model at the viscosity ν dictate the level of uncertainties. The switching control in the SMC law works in favor of keeping the trajectory on the defined sliding surface in the presence of the aforedescribed uncertainties.

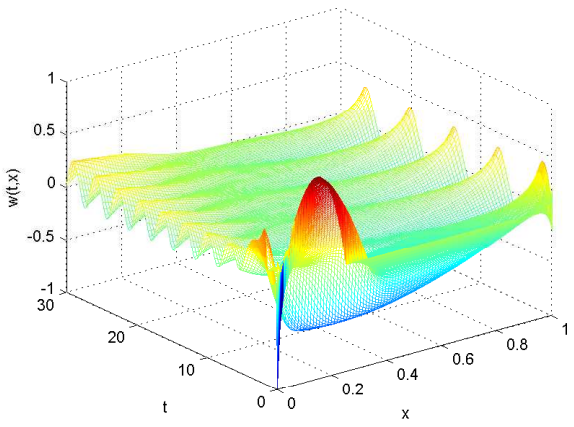


Figure 10.9: Velocity response of the full-order model with the proposed SMC law to track a sinusoidal function.

10.5 Concluding Remarks

In this paper, we derived and validated a reduced-order model for Burgers' equation with Neumann boundary conditions, where the reduced-order model was obtained by utilizing a combination of the POD-Galerkin method and weak solution approach. It was shown that extracting the POD bases associated with the reduced-order model only once at the nominal value of the viscosity parameter ν_0 is a source of uncertainty in the case of varying viscosity. It was proven that the error terms associated with the model reduction and changing viscosity is bounded in terms of the system matrices at the nominal value of viscosity ν_0. The developed reduced-order model was then used for the design of a robust sliding mode controller on the basis of the sliding surfaces defined according to the reduced-order model at ν_0. Due to the need for state measurements or estimates for the sliding mode controller, implementing the calculated reduced model only once at the nominal value of the viscosity significantly decreases the computational load for both controller and the observer design. The reduced-order model-based controller requires the computation of the reduced model for the varying parameter ν; however, the proposed robust sliding mode controller implements the obtained reduced model at ν_0 and guarantees the desired tracking performance of

the full-order model at the new parameter ν. Finally, numerical studies demonstrated promising results by using the proposed reduced-order model and controller to achieve a high performance in reference trajectory tracking. The results proved the practicality of the proposed reduced order modeling and control design approaches for fluid systems governed by Burgers' equation.

CHAPTER 11

CONCLUDING REMARKS

In the present dissertation, a distributed control approach has been designed to handle the scenarios associated with deployment of heterogeneous groups of agents in a given uncertain environment. First, a team-based approach has been developed to divide the region among multiple groups of agents where each team may seek a different objective. The team-based approach is further enhanced by considering the changing boundaries of the teams. The formation control has been proposed within the team framework to fulfill the underlying constraints in the deployment of the agent. The mathematical framework associated with the dynamic boundaries has been used to develop a coverage approach for environments with changing boundaries like rivers. Last but not least, the heterogeneity in the underlying assigned set of tasks has been formulated in deployment of teams of agents when different agents pursue different set of objectives. The developed approaches are implemented and studies through a set of numerical simulations where it is shown that the proposed methods can be successfully used to handle heterogeneity in an uncertain environment.

As the next task in this dissertation, two deterministic and stochastic approaches are proposed for identification of LPV systems. In the deterministic method, the problem has been formulated in a way to yield a solution that can handle errors in the scheduling variables. This allowed the kernel-based identification method to partially compensate for the error in p to avoid misestimating of the system parameters and lead to a set of new expressions for LPV model coefficients by changing the basis functions. On the other hand, the stochastic approach leads to acquiring a better understanding of the uncertainties in data through more accurate formulation of the noise effect on the LPV model coefficients compared to the standard GP. The results indicate that the proposed

method gives a more accurate estimation of the LPV model coefficient functions in the presence of both noisy measurement outputs and erroneous scheduling variables.

The proposed robust nonlinear control approach can effectively cope with the uncertainty in the model parameters. First, a reduced-order model has been developed for a parabolic PDE to capture the dynamics of the full-order model. Then, a reduced-order model based sliding mode controller is designed to ensure the stability of the full-order model. As the next step, a sensitivity analysis is carried out on the effect of the varying parameters on the extracted reduced-order model. The boundedness of the error terms have investigated and a robust controller is designed accordingly. The approach has been evaluated via a numerical example.

11.1 FUTRUE RESEARCH DIRECTIONS

Development of robust and stochastic techniques for system identification and control of nonlinear systems is of great importance with increasing use of such approaches in real world applications. Although, it has been attempted to focus on three major areas of uncertain nonlinear systems throughout this dissertation, there are many more open problems that can be addressed as the future direction of this research. Possible future directions include:

- To enhance the identification results by introducing a more accurate estimation of the effect of uncertain scheduling variables rather than the linear estimation.
- To develop a robust technique by assumption of noisy measurements used for extracting the reduced-order model.
- To implement the multi-tasking within the team framework.
- To develop a performance based partitioning method, i.e., power diagrams, to get a more realistic coverage approach.

BIBLIOGRAPHY

[1] Distributed coverage control for concave areas by a heterogeneous robotswarm with visibility sensing constraints. *Automatica*, 53:195 – 207, 2015.

[2] F. Abbasi, A. Mesbahi, and J. Mohammadpour. Team-based coverage control of moving sensor networks. In *2016 American Control Conference (ACC)*, pages 5691–5696, July 2016.

[3] F. Abbasi, A. Mesbahi, and J. Mohammadpour. Team-based coverage control of moving sensor networks. In *American Control Conference (ACC)*, July. 2016, to appear.

[4] Farshid Abbasi and Javad Mohammadpour. Nonlinear model order reduction of burgers' equation using proper orthogonal decomposition. In *2015 American Control Conference (ACC)*, pages 583–588. IEEE, 2015.

[5] Masoud Abbaszadeh and Horacio J Marquez. Robust state observation for sampled-data nonlinear systems with exact and euler approximate models. In *2007 American Control Conference*, pages 1687–1692. IEEE, 2007.

[6] Remi Abgrall and David Amsallem. Robust model reduction by L^1-norm minimization and approximation via dictionaries: Application to linear and nonlinear hyperbolic problems. *arXiv preprint arXiv:1506.06178*, 2015.

[7] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh. Self-supervised segmentation of river scenes. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 6227–6232, May 2011.

[8] Cyrille Allery, Aziz Hamdouni, David Ryckelynck, and N Verdon. A priori reduction method for solving the two-dimensional burgers equations. *Applied Mathematics and Computation*, 217(15):6671–6679, 2011.

[9] David Amsallem and Charbel Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.

[10] Gökhan M Atınç, Dušan M Stipanović, and Petros G Voulgaris. Supervised coverage control of multi-agent systems. *Automatica*, 50(11):2936–2942, 2014.

[11] Franz Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[12] E. Bakolas. Partitioning algorithms for homogeneous multi-vehicle systems with planar rigid body dynamics. In *Decision and Control (CDC), IEEE 53rd Annual Conference on*, pages 5393–5398, Dec 2014.

[13] B. Bamieh and L. Giarre. Identification of linear parameter varying models. *Int. J. Robust Nonlinear Control*, 12(9):841–853, 2002.

[14] Opher Baron, Oded Berman, Dmitry Krass, and Qian Wang. The equitable location problem on the plane. *European Journal of Operational Research*, 183(2):578–590, 2007.

[15] Ulrike Baur, Peter Benner, and Lihong Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Archives of Computational Methods in Engineering*, 21(4):331–358, 2014.

[16] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[17] Peter Benner, Volker Mehrmann, and Danny C Sorensen. *Dimension reduction of large-scale systems*, volume 45. Springer, 2005.

[18] Peter Benner, Ekkehard Sachs, and Stefan Volkwein. *Model order reduction for PDE constrained optimization*. Springer, 2014.

[19] David Betounes. *Partial differential equations for computational science: with Maple and vector analysis*, volume 1. Springer Science & Business Media, 1998.

[20] Jean-Marie Buchot, Jean-Pierre Raymond, and Jorge Tiago. Coupling estimation and control for a two dimensional Burgers type equation. *ESAIM: Control, Optimisation and Calculus of Variations*, 21(2):535–560, 2015.

[21] Tan Bui-Thanh, Murali Damodaran, and Karen E Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal*, 42(8):1505–1516, 2004.

[22] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[23] Mark Butcher, Alireza Karimi, and Roland Longchamp. On the consistency of certain identification methods for linear parameter varying systems. *IFAC Proceedings Volumes*, 41(2):4018–4023, 2008.

[24] V Cerone, D Piga, and D Regruto. Set-membership identification of input-output lpv models with uncertain time-varying parameters. *Linear parameter-varying system identification: new developments and trends*, 14:41, 2012.

[25] Vito Cerone, Dario Piga, and Diego Regruto. A convex relaxation approach to set-membership identification of lpv systems. *Automatica*, 49(9):2853–2859, 2013.

[26] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh. Perception for a river mapping robot. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 227–234, Sept 2011.

[27] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*, 11(Apr):1471–1490, 2010.

[28] Jian Chen, Yong Guan, and U. Pooch. Customizing GPSR for wireless sensor networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 549–551, Oct. 2004.

[29] Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[30] A. Chiuso and G. Pillonetto. Bayesian and nonparametric methods for system identification and model selection. In *Control Conference (ECC), 2014 European*, pages 2376–2381, June 2014.

[31] Kai Lai Chung. *A course in probability theory*. Academic press, 2001.

[32] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, April 2004.

[33] Patrick Dallaire, Camille Besse, and Brahim Chaib-Draa. Learning gaussian process models from uncertain data. In *International Conference on Neural Information Processing*, pages 433–440. Springer, 2009.

[34] L. DeVries and D.A. Paley. Multivehicle control in a strong flowfield with application to hurricane sampling. *Journal of Guidance, Control, and Dynamics*, 35(3):794–806, 2012.

[35] Feng Ding, Yang Shi, and Tongwen Chen. Auxiliary model-based least-squares identification methods for hammerstein output-error systems. *Systems & Control Letters*, 56(5):373–380, 2007.

[36] Sergey V Drakunov and Vadim I Utkin. Sliding mode control in dynamic systems. *International Journal of Control*, 55(4):1029–1037, 1992.

[37] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev.*, 41(4):637–676, 1999.

[38] Stevan Dubljevic, Nael H El-Farra, Prashant Mhaskar, and Panagiotis D Christofides. Predictive control of parabolic PDEs with state and control constraints. *International Journal of robust and nonlinear control*, 16(16):749–772, 2006.

[39] M. Dunbabin and A. Grinham. Experimental evaluation of an autonomous surface vehicle for water quality and greenhouse gas emission monitoring. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5268–5274, May 2010.

[40] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics Automation Magazine*, 19(1):24–39, March 2012.

[41] Matthew Dunbabin, Alistair Grinham, and James Udy. An autonomous surface vehicle for water quality monitoring. In *Australasian Conference on Robotics and Automation (ACRA)*, pages 2–4, Dec 2009.

[42] Kenneth Eriksson. *Computational differential equations*, volume 1. Cambridge University Press, 1996.

[43] Lawrence C Evans. *Partial differential equations*. American Mathematical Society, 2002.

[44] Clive Fletcher. *Computational techniques for fluid dynamics 2: Specific techniques for different flow categories*. Springer Science & Business Media, 2012.

[45] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[46] M. Garey, D. Johnson, and H. Witsenhausen. The complexity of the generalized Lloyd - max problem (corresp.). *IEEE Transactions on Information Theory*, 28(2):255–256, 1982.

[47] Agathe Girard, Carl Edward Rasmussen, Joaquin Quinonero-Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs: application to multiple-step ahead time series forecasting. 2003.

[48] Ivan Goethals, Kristiaan Pelckmans, Johan AK Suykens, and Bart De Moor. Identification of mimo hammerstein models using least squares support vector machines. *Automatica*, 41(7):1263–1272, 2005.

[49] Arash Golabi, Nader Meskin, Roland Tóth, and Javad Mohammadpour. A bayesian approach for estimation of linear-regression lpv models. In *53rd IEEE Conference on Decision and Control*, pages 2555–2560. IEEE, 2014.

[50] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance. *IEEE Robotics Automation Magazine*, 13(3):16–25, Sept 2006.

[51] Christian Grossmann, Hans-Görg Roos, and Martin Stynes. *Numerical treatment of partial differential equations*. Springer, 2007.

[52] Chun-li Guo, Cheng-kang Xie, and C Zhou. Stabilization of a spatially non-causal reaction–diffusion equation by boundary control. *International Journal of Robust and Nonlinear Control*, 24(1):1–17, 2014.

[53] M. Amac Guvensan and A. Gokhan Yavuz. On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks*, 9(7):1238 – 1255, 2011.

[54] J. Habibi, H. Mahboubi, and A. Aghdam. Distributed coverage control of mobile sensor networks subject to measurement error. *IEEE Transactions on Automatic Control*, 2016, to appear.

[55] J. Habibi, H. Mahboubi, and A. G. Aghdam. A gradient-based coverage optimization strategy for mobile sensor networks. *IEEE Transactions on Control of Network Systems*, 2016, to appear.

[56] Alexander Hay, Imran Akhtar, and Jeff T Borggaard. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 68(1):122–134, 2012.

[57] Alexander Hay, Jeff Borggaard, Imran Akhtar, and Dominique Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of computational Physics*, 229(4):1327–1352, 2010.

[58] Alexander Hay, Jeffrey T Borggaard, and Dominique Pelletier. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2009.

[59] Matthias Heinkenschloss. *Numerical solution of implicitly constrained optimization problems*. Citeseer, 2008.

[60] Dietmar Hömberg and Stefan Volkwein. Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition. *Mathematical and computer modelling*, 38(10):1003–1028, 2003.

[61] Mei Hong, Torsten Söderström, and Wei Xing Zheng. Accuracy analysis of bias-eliminating least squares estimates for errors-in-variables systems. *Automatica*, 43(9):1590–1596, 2007.

[62] Saeid Jafari, Amir Ajorlou, and Amir G Aghdam. Leader localization in multi-agent systems subject to failure: A graph-theoretic approach. *Automatica*, 47(8):1744–1750, 2011.

[63] Sezal Jain, Stephen Nuske, Andrew Chambers, Luke Yoder, Hugh Cover, Lyle Chamberlain, Sebastian Scherer, and Sanjiv Singh. *Field and Service Robotics: Results of the 9th International Conference*, chapter Autonomous River Exploration, pages 93–106. Springer International Publishing, Cham, 2015.

[64] Martin Kahlbacher and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discussiones Mathematicae, Differential Inclusions, Control and Optimization*, 27(1):95–117, 2007.

[65] Yiannis Kantaros, Michalis Thanou, and Anthony Tzes. Distributed coverage control for concave areas by a heterogeneous robot–swarm with visibility sensing constraints. *Automatica*, 53:195–207, 2015.

[66] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische mathematik*, 90(1):117–148, 2001.

[67] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515, 2002.

[68] Karl Kunisch, Stefan Volkwein, and L Xie. Hjb-pod-based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, 3(4):701–722, 2004.

[69] A. Kwok and S. Martinez. A coverage algorithm for drifters in a river environment. In *Proceedings of the 2010 American Control Conference*, pages 6436–6441, June 2010.

[70] A. Kwok and S. Martinez. Deployment of drifters in a piecewise-constant flow environment. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6584–6589, Dec 2010.

[71] Andrew Kwok and Sonia Martinez. Coverage maximization with autonomous agents in fast flow environments. *Journal of Optimization Theory and Applications*, 155(3):986–1007, 2012.

[72] Oliver Lass and Stefan Volkwein. Pod galerkin schemes for nonlinear elliptic-parabolic systems. *SIAM Journal on Scientific Computing*, 35(3):A1271–A1298, 2013.

[73] Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for modeling and computational reduction*, pages 235–273. Springer, 2014.

[74] V. Laurain, W. X. Zheng, and R. Tth. Introducing instrumental variables in the ls-svm based identification framework. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3198–3203, Dec 2011.

[75] Vincent Laurain, Roland Toth, Wei Xing Zheng, and Marion Gilson. Nonparametric identification of lpv models under general noise conditions: an ls-svm based approach. *IFAC Proceedings Volumes*, 45(16):1761–1766, 2012.

[76] Eun-Taik Lee and Hee-Chang Eun. Damage identification based on the proper orthogonal mode energy curvature. *Journal of Vibration and Acoustics*, 137(4):041018, 2015.

[77] H. C. Lee, C. Y. Cho, C. T. King, Y. M. Fang, and B. J. Lee. Design and implementation of non-autonomous mobile wireless sensor for debris flow monitoring. In *2009 IEEE 6th International Conference on Mobile Ad-hoc and Sensor Systems*, pages 1062–1064, Oct 2009.

[78] S.G. Lee, Y. Diaz-Mercado, and M. Egerstedt. Multirobot control using time-varying density functions. *IEEE Transactions on Robotics*, 31(2):489–493, April 2015.

[79] Philippe Lemmerling, Nicola Mastronardi, and Sabine Van Huffel. Fast algorithm for solving the hankel/toeplitz structured total least squares problem. *Numerical Algorithms*, 23(4):371–392, 2000.

[80] Shuai Li, Ruofan Kong, and Yi Guo. Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Transactions on Mechatronics*, 19(6):1810–1820, Dec 2014.

[81] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent and Robotic Systems*, 72(2):147–165, 2013.

[82] Lennart Ljung. *System Identification*, pages 163–173. Birkhäuser Boston, Boston, MA, 1998.

[83] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, Mar 1982.

[84] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[85] A. Macwan, G. Nejat, and B. Benhabib. Optimal deployment of robotic teams for autonomous wilderness search and rescue. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4544–4549, Sept. 2011.

[86] Christopher S Magirl, Jeffrey W Gartner, Graeme M Smart, and Robert H Webb. Water velocity and the nature of critical flow in large rapids on the Colorado river, Utah. *Water Resources Research*, 45(5):454–459, 2009.

[87] H. Mahboubi, K. Moezzi, A. G. Aghdam, and K. Sayrafian-Pour. Distributed deployment algorithms for efficient coverage in a network of mobile sensors with nonidentical sensing capabilities. *IEEE Transactions on Vehicular Technology*, 63(8):3998–4016, Oct 2014.

[88] Ivan Markovsky and Sabine Van Huffel. Overview of total least-squares methods. *Signal processing*, 87(10):2283–2302, 2007.

[89] Fabio Morbidi, Francesco Bullo, and Domenico Prattichizzo. Visibility maintenance via controlled invariance for leader–follower vehicle formations. *Automatica*, 47(5):1060–1067, 2011.

[90] Swathi M Mula and Charles E Tinney. A study of the turbulence within a spiralling vortex filament using proper orthogonal decomposition. *Journal of Fluid Mechanics*, 769:570–589, 2015.

[91] K-R Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, and Bernhard Scholkopf. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, 12(2):181–201, 2001.

[92] Tetsushi Nishida and Kokichi Sugihara. *Algorithms and Computation: 14th International Symposium, ISAAC 2003, Kyoto, Japan, December 15-17, 2003. Proceedings*, chapter Voronoi Diagram in the Flow Field, pages 26–35. Springer, Berlin Heidelberg, 2003.

[93] Tetsushi Nishida, Kokichi Sugihara, and Masato Kimura. Stable marker-particle method for the voronoi diagram in a flow field. *Journal of Computational and Applied Mathematics*, 202(2):377 – 391, 2007.

[94] Ahmed K Noor. Recent advances in reduction methods for nonlinear problems. *Computers & Structures*, 13(1):31–44, 1981.

[95] Ahmed K Noor and Jeanne M Peters. Reduced basis technique for nonlinear analysis of structures. *Aiaa journal*, 18(4):455–462, 1980.

[96] Ahmed K Noor, JM Peters, and CM ANDERSEN. Reduced basis technique for collapse analysis of shells. *AIAA Journal*, 19(3):393–397, 1981.

[97] Cameron Nowzari and Jorge Cortés. Self-triggered coordination of robotic networks for optimal deployment. *Automatica*, 48(6):1077–1087, 2012.

[98] Stephen Nuske, Sanjiban Choudhury, Sezal Jain, Andrew Chambers, Luke Yoder, Sebastian Scherer, Lyle Chamberlain, Hugh Cover, and Sanjiv Singh. Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers. *Journal of Field Robotics*, 32(8):1141–1162, 2015.

[99] J. Le Ny and G. J. Pappas. Adaptive deployment of mobile robotic networks. *IEEE Transactions on Automatic Control*, 58(3):654–666, March 2013.

[100] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[101] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Inc., New York, NY, USA, 1992.

[102] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Inc., New York, NY, USA, 1992.

[103] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, March 2006.

[104] Mehmet Önder Efe and Hİtay Özbay. Low dimensional modeling and boundary controller design for Burgers equation. *International Journal of Control*, 77(10):895–906, 2004.

[105] Prof. David A. Lowther Oszkár Bíró, Prof Prof. Piergiorgio Alotto, Stefan Volkwein, and Andrea Wesche. The reduced basis method applied to transport equations of a lithium-ion battery. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 32(5):1760–1772, 2013.

[106] Prof. David A. Lowther Oszkár Bíró, Prof Prof. Piergiorgio Alotto, Stefan Volkwein, and Andrea Wesche. The reduced basis method applied to transport equations of a lithium-ion battery. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 32(5):1760–1772, 2013.

[107] Ying Ouyang. Evaluation of river water quality monitoring stations by principal component analysis. *Water Research*, 39(12):2621 – 2635, 2005.

[108] HM Park and DH Cho. The use of the karhunen-loeve decomposition for the modeling of distributed parameter systems. *Chemical Engineering Science*, 51(1):81–98, 1996.

[109] R. Patel, P. Frasca, and F. Bullo. Centroidal area-constrained partitioning for robotic networks. *Journal of Dynamic Systems, Measurement, and Control*, 136(3):137–156, 2013.

[110] R. Patel, P. Frasca, J. W. Durham, R. Carli, and F. Bullo. Dynamic partitioning and coverage control with asynchronous one-to-base-station communication. *IEEE Transactions on Control of Network Systems*, 3(1):24–33, March 2016.

[111] M. Pavone, E. Frazzoli, and F. Bullo. Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Transactions on Automatic Control*, 56(6):1259–1274, June 2011.

[112] Wilfrid Perruquetti and Jean-Pierre Barbot. *Sliding mode control in engineering*. CRC Press, 2002.

[113] Dario Piga, Pepijn Cox, Roland Tóth, and Vincent Laurain. Lpv system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53:329–338, 2015.

[114] G Pillonetto and A Chiuso. Tuning complexity in kernel-based linear system identification: The robustness of the marginal likelihood estimator. In *Control Conference (ECC), 2014 European*, pages 2386–2391. IEEE, 2014.

[115] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

[116] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Decision and Control, 47th IEEE Conference on*, pages 3947–3952, Dec 2008.

[117] Joaquin Quiñonero-Candela and Sam T Roweis. Data imputation and robust training with gaussian processes. 2003.

[118] Carl Edward Rasmussen. *Evaluation of Gaussian processes and other methods for non-linear regression*. PhD thesis, University of Toronto, Canada, 1996.

[119] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[120] Syed Z Rizvi, Javad Mohammadpour, Roland Tóth, and Nader Meskin. Parameter set-mapping using kernel-based pca for linear parameter-varying systems. In *Control Conference (ECC), 2014 European*, pages 2744–2749. IEEE, 2014.

[121] J Ben Rosen, Haesun Park, and John Glick. Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Analysis and Applications*, 17(1):110–126, 1996.

[122] Clarence W Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.

[123] Yu Ru and Sonia Martinez. Coverage control in constant flow environments based on a mixed energytime metric. *Automatica*, 49(9):2632 – 2640, 2013.

[124] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *Int. J. Rob. Res.*, 28(3):357–375, 2009.

[125] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[126] F. Sharifi, A. Chamseddine, H. Mahboubi, Youmin Zhang, and A.G. Aghdam. A distributed deployment strategy for a network of cooperative autonomous vehicles. *IEEE Transactions on Control Systems Technology*, 23(2):737–745, March 2015.

[127] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding mode control and observation*. Springer, 2014.

[128] V. Simeonov, J. Einax, I. Stanimirova, and J. Kraft. Environmetric modeling and interpretation of river water monitoring data. *Analytical and Bioanalytical Chemistry*, 374(5):898–905, 2002.

[129] Lawrence Sirovich. Turbulence and the dynamics of coherent structures part i: coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[130] Lyle C Smith III. *Finite element Approximations of Burgers' Equation with Robin's Boundary Conditions*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[131] Alex J Smola and Bernhard Schölkopf. *Learning with kernels*. Citeseer, 1998.

[132] Torsten Söderström. Errors-in-variables methods in system identification. *Automatica*, 43(6):939–958, 2007.

[133] Ercan Solak, Roderick Murray-Smith, William E Leithead, Douglas J Leith, and Carl Edward Rasmussen. Derivative observations in gaussian process models of dynamic systems. 2003.

[134] Cheng Song, Lu Liu, Gang Feng, and Shengyuan Xu. Coverage control for heterogeneous mobile sensor networks on a circle. *Automatica*, 63:349–358, 2016.

[135] Yiannis Stergiopoulos and Anthony Tzes. Spatially distributed area coverage optimisation in mobile robotic networks with arbitrary convex anisotropic patterns. *Automatica*, 49(1):232–237, 2013.

[136] Housheng Su, Xiaofan Wang, and Zongli Lin. Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, 2009.

[137] Kokichi Sugihara. Voronoi diagrams in a river. *International Journal of Computational Geometry & Applications*, 2(1):29–48, 1992.

[138] M. Sznaier, C. Mazzaro, and T. Inanc. An lmi approach to control oriented identification of lpv systems. In *American Control Conference, 2000. Proceedings of the 2000*, volume 5, pages 3682–3686 vol.5, 2000.

[139] P. Tabuada, G. J. Pappas, and P. Lima. Motion feasibility of multi-agent formations. *IEEE Transactions on Robotics*, 21(3):387–392, June 2005.

[140] H. G. Tanner and D. K. Christodoulakis. Cooperation between aerial and ground vehicle groups for reconnaissance missions. In *Decision and Control, 45th IEEE Conference on*, pages 5918–5923, Dec 2006.

[141] D. Tao and T. Y. Wu. A survey on barrier coverage problem in directional sensor networks. *IEEE Sensors Journal*, 15(2):876–885, Feb 2015.

[142] F. Tauro, G. Olivieri, A. Petroselli, M. Porfiri, and S. Grimaldi. Flow monitoring with a camera: a case study on a flood event in the tiber river. *Environmental Monitoring and Assessment*, 188(2):1–11, 2016.

[143] O. Tekdas and V. Isler. Sensor placement algorithms for triangulation based localization. In *Robotics and Automation, IEEE International Conference on*, pages 4448–4453, April 2007.

[144] F. Thompson, D. Gilvear, A. Tree, and R. Jeffries. Quantification of freshwater pearl mussel entrainment velocities and controlling factors; a flume study. *River Research and Applications*, 2015, to appear.

[145] Roland Tóth. *Modeling and identification of linear parameter-varying systems*, volume 403. Springer, 2010.

[146] R. Tth, V. Laurain, W. X. Zheng, and K. Poolla. Model structure learning: A support vector machine approach for lpv linear-regression models. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3192–3197, Dec 2011.

[147] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[148] Veerle Verschoren, Dieter Meire, Jonas Schoelynck, Kerst Buis, Kris D. Bal, Peter Troch, Patrick Meire, and Stijn Temmerman. Resistance and reconfiguration of natural flexible submerged vegetation in hydrodynamic river modelling. *Environmental Fluid Mechanics*, 16(1):245–265, 2015.

[149] Stefan Volkwein. Distributed control problems for the burgers equation. *Computational Optimization and Applications*, 18(2):115–140, 2001.

[150] Stefan Volkwein. Proper orthogonal decomposition: applications in optimization and control. *CEA-EDFINRIA Numerical Analysis Summer School*, 2007.

[151] A. Wallar, E. Plaku, and D.A. Sofge. Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas. *IEEE Transactions on Automation Science and Engineering*, 12(3):969–980, July 2015.

[152] Zhiguang Xiong, Yanping Chen, and Yan Zhang. Convergence of FEM with interpolated coefficients for semilinear hyperbolic equation. *Journal of Computational and Applied mathematics*, 214(1):313–317, 2008.

[153] Zaihua Xu and Yungang Liu. Adaptive boundary stabilization for first-order hyperbolic PDEs with unknown spatially varying parameter. *International Journal of Robust and Nonlinear Control*, 26(3):613–628, 2016.

[154] Wei Xing Zheng. Transfer function estimation from noisy input and output data. *International Journal of Adaptive Control and Signal Processing*, 12(4):365–380, 1998.

[155] Y. Zheng, Y. Zhu, and L. Wang. Consensus of heterogeneous multi-agent systems. *IET Control Theory Applications*, 5(16):1881–1888, Nov. 2011.