GAIN-SCHEDULED CONTROL BASED ON ONLINE ESTIMATED SENSOR AGING

By

Aqeel Madhag

A DISSERTATION

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

Electrical Engineering - Doctor of Philosophy

2019

ABSTRACT

GAIN-SCHEDULED CONTROL BASED ON ONLINE ESTIMATED SENSOR AGING

By

Aqeel Madhag

This work proposes that sensor performance degradation and sensor sudden failure due to its aging can be characterized by the sensor measurement noise covariance variation (shift), while recent literature considers sensor fault and/or failure as an augmented state. That is, in this work gradual sensor performance degradation due to its aging can be characterized by gradual-variation of the time-varying sensor measurement noise covariance. In addition, sensor abrupt or intermittent fault or failure can be characterized by an intermittent or abrupt change of the time-varying sensor noise covariance. Furthermore, this work proposes fault detection algorithms to online monitoring sensor performance and online detecting and identifying sensor performance degradation and sudden (abrupt or intermittent) failure due to sensor aging. The proposed algorithms have two key features: online estimating the slowly-varying sensor measurement noise covariance and detecting the sudden (fast) change of the sensor measurement noise covariance. The first proposed algorithm shows the capability of estimating the slowly-varying sensor measurement noise covariance for multiple-input and multiple-output systems with time-varying sensor measurement noise covariance. Furthermore, the proposed estimation algorithm shows a reasonable rate of convergence, better estimation accuracy and less computation load in contrast to published literature. Moreover, the second proposed algorithm, which is a memory-based technique calculating the Euclidean distance of estimated covariance matrices between two sliding estimation windows, is used to detect the abrupt (or intermittent) change of sensor noise covariance matrix. The proposed algorithm ori-

ginally is designed for discrete linear time-varying (DLTV) systems and applied to discrete linear parameter-varying (DLPV) systems. The proposed algorithm shows the capability of detecting the abrupt (or intermittent) change of sensor measurement noise covariance for multiple-input and multiple-output discrete linear parameter-varying systems with time-varying sensor measurement noise covariance, where the scheduling parameters lie within a compact set. Furthermore, the proposed estimation algorithm shows a reasonable rate of convergence, better estimation accuracy and less computation load in contrast to published literature. The other major contribution of this work is the characterization of the control synthesis conditions using parametrized linear matrix inequalities (PLMI) for a multi-objective gain-scheduled noisy output-feedback controller that minimizes the output cost on $\mathcal{H}_2$ performance with satisfactory system stability, $\mathcal{H}_\infty$ performance and control input covariance constraints ($\mathcal{H}_2$ constraints on the control inputs) in the presence of sensor aging. The closed-loop system stability and performance, in terms of mixed $\mathcal{H}_2/\mathcal{H}_\infty$ performances, relative improvement, numerical complexity, computation time, and initial conditions response are studied. The synthesized controller guarantees not only the stability but also the closed-loop mixed $\mathcal{H}_2/\mathcal{H}_\infty$ performances and it is feasible for real-time applications. To generate the output system performance, the output covariance constraints (OCC) control synthesis conditions are developed using parametrized linear matrix inequalities (PLMI) for a gain-scheduled noisy output-feedback controller that minimizes the cost on control input (control effort) with satisfactory system output covariance constraints in the presence of sensor aging. The closed-loop system performance in terms of control effort as a function of the output covariance and the sensor noise covariance is studied. The synthesized controller guarantees the closed-loop OCC performance and it is feasible for real-time applications. The synthesized control utilizes sensor aging information to minimize its effect on the system and improves the closed-loop system performance as possible subject to given constraints and sensor performance degradation due to aging.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Guoming Zhu for the continuous support through all my Ph.D. research, for his guidance, advise, motivation, and immense knowledge. His guidance helped me in all the time of research. I could not have imagined having a better advisor and mentor for my Ph.D study. In my opinion, I will never ever meet a person has sweet heart like him.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Hassen Khalil, Prof. Ranjan Mukerjee, and Prof. Vaibhav Srivastava, for their insightful comments, which incented me to widen my research from various perspectives.

I thank my wife in for supporting me spiritually throughout the whole Ph.D. journey, for the sleepless nights I were working, for taking care of our family, for her patience when I were lost, for her love, and for all the fun we have had in the past years. In addition, I thank my kids for their love and smiles which give me the power to work hard.

My sincere appreciation also goes to prof. Tim Hogan, who provided me an opportunity to be a good instructor. Without his precious support it would not be possible to improve my teaching skills.

Last but not the least, I would like to thank everyone who believed in me and supported me spiritually throughout my Ph.D. and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO SYMBOLS

$\omega(k)$	H_2 process noise
$\nu(k)$	H_2 sensor measurement noise
$\mathbf{W}_k$	time-varying process matrix
$\mathbf{V}_k$	time-varying measurement noise covariance matrix
$\hat{\mathbf{V}}_k$	estimated time-varying measurement noise covariance matrix
$\tilde{\mathbf{v}}_k$	innovation vector
φ	forgetting factor
ϵ	scalar constant
ε_k	sum of square error
N	estimation window size
m	sliding estimation window size
$\mathcal{I}_{i,j}$	indicator function (signal)
RI	relative improvement
η	H_∞ performance upper bound
χ	H_2 performance upper bound

KEY TO ABBREVIATIONS

FTC This is a Fault Tolerant Control

LPV This is a Linear Parameter Varying

LTV This is a Linear Time-Varying

LTI This is a Linear Time-Invariant

DLTV This is a Discrete Linear Time-Varying

DLPV This is a Discrete Linear Parameter Varying

SOF This is a Static Output-Feedback

DOF This is a Dynamic Output-Feedback

FGS This is a Full Gain-Scheduled

PGS This is a Partial Gain-Scheduled

KF This is a Kalman Filter

AKF This is a Adaptive Kalman Filter

MIMO This is a Multiple-Input Multiple-Output

LMI This is a Linear Matrix Inequality

PLMI This is a Parametrized LMI

ICC This is a Input Covariance Constraint

GSNOF This is a Gain-Scheduled Noisy static Output Feedback

Chapter 1

Introduction

As technological systems become more complex, the requirement for dependable and repeatable system performance is very important for modern control systems heavily relying on sensor signals for feedback control. This is particularly important for safety-critical applications, where a successful mission of protecting human life, property, and/or environment becomes a paramount goal. To minimize the possibility of unexpected failures, control systems need to have increased reliability. One way to improve the system reliability is to enhance its ability of fault detection. There are many system failure sources, and the most common and significant one is sensor failure. To be specific, sensor performance degradation and/or fault due to sensor aging. Indeed, in control systems, the sensor (or set of sensors) plays (play) the rule of an interpreter between physical plant from one side and the online world and controller from another side; see Fig 1.1 for more details. That is, sensor fault or failure affects the system output performance directly and could lead to catastrophic consequences. Therefore, a faulty sensor may cause system performance degradation, system shut down, and/or fatal accident. On the other hand, designing a controller not consider sensor aging information may cause system performance degradation, system shut down, and/or fatal accident. Motivated by this problem, the main contributions of this work are as follow:

- This work assumes that sensor performance shift (i.e., sensor performance gradual-degradation and/or fault due to sensor aging) can be characterized by its measurement noise covariance shift. That is, the gradual-degradation of sensor performance due to its aging is charac-

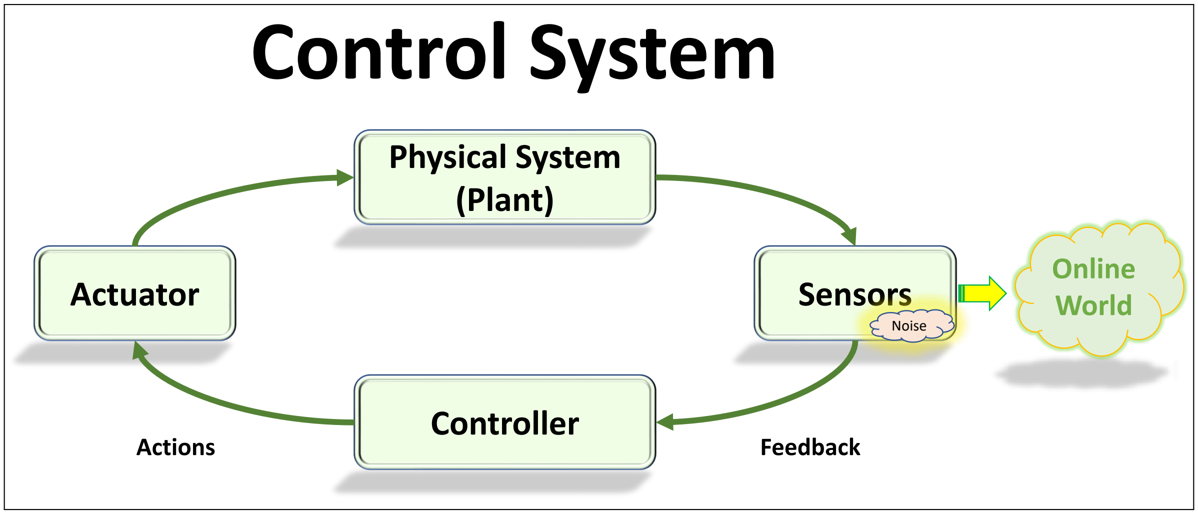


Figure 1.1: General control system.

terized by the slowly-varying sensor measurement noise covariance, and the sensor abrupt (or intermittent) failures are characterized by an abrupt (or intermittent) change of sensor measurement noise covariance.

- An algorithm is proposed to detect gradual sensor performance degradation due to sensor aging that is characterized by slowly-varying sensor measurement noise covariance. In addition, another algorithm is proposed to detect and identify the sudden (abrupt or intermittent) sensor failure, assuming that the sensor performance sudden shift can be characterized by the noise covariance sudden (abrupt or intermittent) variation of the sensor measurement.

- Synthesis multi-objective gain-scheduled noisy output-feedback (GSNOF) controller compensates for the gradually-degraded sensor performance due to sensor aging and maintains the desired performance. To be specific, a multi-objective gain-scheduled noisy output-feedback controller (GSNOF), utilizing sensor aging information, such that system stability and desired performances are guaranteed with optimal output covariance performance subject to constraints on the control input covariance matrix (ICC) and $\mathcal{H}_\infty$ performance is synthesized. In addition, another controller is synthesized. That is, synthesis of a gain-scheduled controller

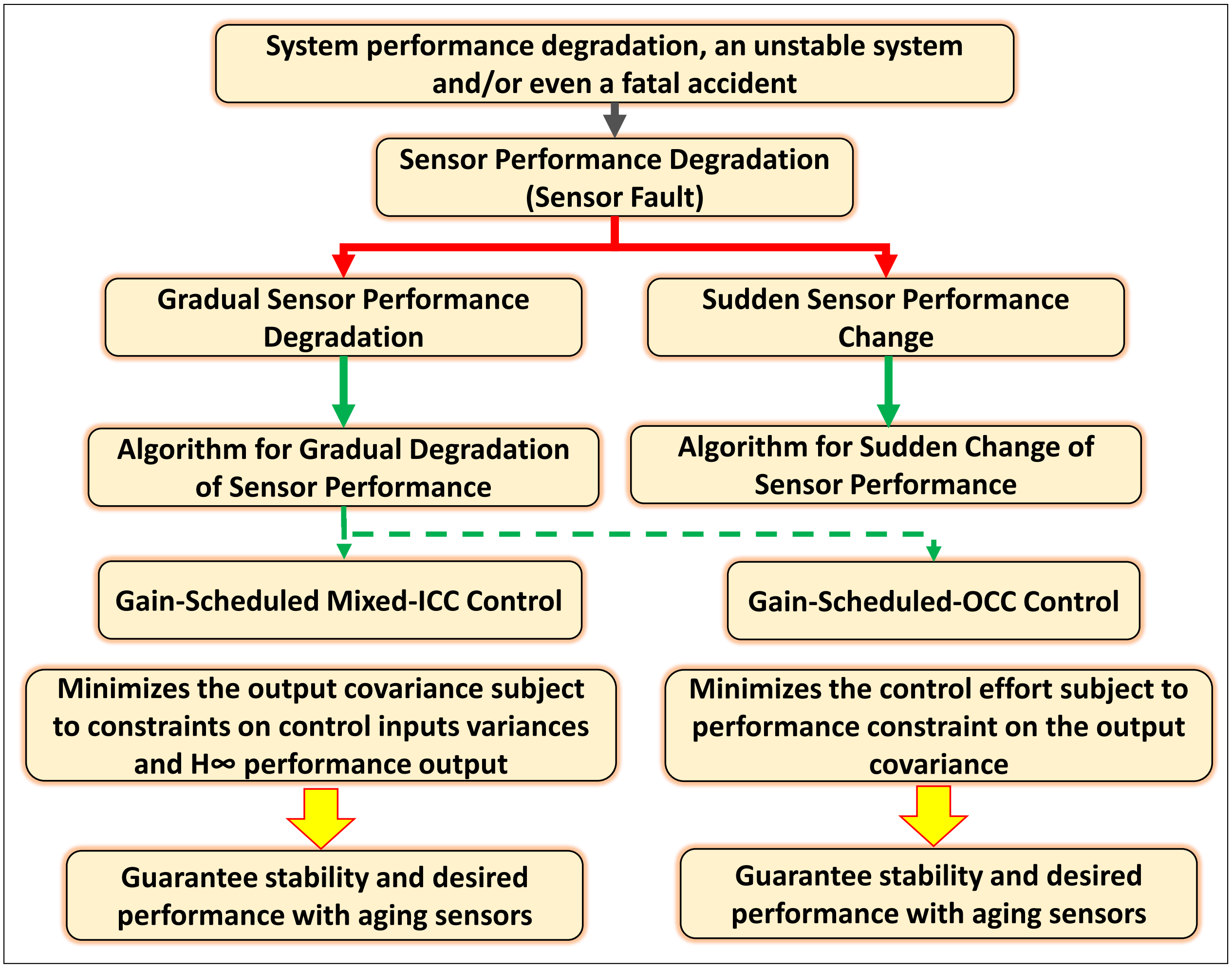


Figure 1.2: The proposed work main contribution parts.

minimizes the control effort subject to performance constraint on output covariance matrix (OCC) in the presence of gradual sensor performance degradation due to sensor aging.

Figure 1.2 shows the itemized main contributions of this work, where the sensor performance degradation information due to sensor aging utilized by the gain-scheduled control to maintain the desired performance (if possible) and satisfy the given constraints, while designing control using conventional methods may lead to system degradation performance or shutdown. Therefore, the proposed sensor fault detection algorithms and the synthesized control give a decision-making technique to keep accept sensor measurements or discard them. The following sub-sections briefly present the motivation and the main contributions of each chapter of this work starting with Chapter 3 since Chapter 2 does not present any theoretical contributions but it is used to present notations, terminologies, and definitions that are used throughout this dissertation..

1.1 Chapter 3 Summary

Chapter 3 presents the proposed algorithm for detecting gradual sensor performance degradation due to sensor aging, assuming that sensor performance shift characterized by its measurement noise covariance shift. In contrast to recent literature, model the sensor fault due to aging as stationary augmented state, this work assumes that the sensor performance gradual-shift (i.e., sensor performance gradual-degradation due to sensor aging) can be characterized by the slowly-varying sensor measurement noise covariance, comparing with the most of the noise statistics identification literature for the case with time-invariant sensor noise covariance only. To match with the goal of online monitoring the sensor performance degradation due to sensor aging, an algorithm is developed (in this chapter) to detect sensor performance degradation due to sensor aging, that is, online estimating slowly-varying sensor measurement noise covariance. To be specific, the proposed algorithm utilizes the covariance-matching technique, along with the adaptive Kalman filter, based on the information about the quality of the weighted innovation sequence to estimate the slowly-varying sensor measurement noise covariance. The covariance-matching of the weighted innovation sequence improves the prediction accuracy and reduces the computational load, making it suitable for real-time applications. The low computation load and online applicability are the primary objectives of the proposed algorithm to online monitor sensor performance and detect the gradual sensor performance degradation due to sensor aging.

1.2 Chapter 4 Summary

This chapter presents an algorithm that used to detect and identify the sudden (abrupt or intermittent) change in sensor performance. In contrast to others, this work assumes that the sensor performance sudden-shift can be characterized by its measurement noise covariance sudden-variation. The proposed algorithm, which is a memory-based technique calculating the Euclidean distance of estimated covariance matrices between two sliding estimation windows, is used to detect the abrupt (or intermittent) change of sensor noise covariance matrix. The memory-based technique is adopted due to its simplicity and online applicability. The iterative manner, without making any assumption on data statistical distribution before and after the sensor performance change, of the proposed algorithm leads to a significant reduction of the computational load, reduced sensitivity to initial conditions and improved estimation accuracy, making it suitable for online applications. An algorithm is proposed to detect and identify the abrupt (or intermittent) sensor failure for discrete linear time-varying (DLTV) systems with application to discrete linear parameter-varying (DLPV) systems as a special case.

1.3 Chapter 5 Summary

This chapter presents the designed multi-objective gain-scheduled controller, with guaranteed stability, that utilizes sensor aging information to minimize the output covariance cost ($\mathcal{H}_2$ norm) function subject to the multiple input covariance constraints (ICC) and $\mathcal{H}_\infty$ performance constraint in the presence of gradual sensor performance degradation due to sensor aging. When closed-loop controllers are designed using the conventional methods, these actuator constraints are usually not taken into account. Consequently, it is possible to design a controller that commands more control power than its capability. On the other hand, with the ICC control design, the actuator constraints are considered during the controller synthesis process. The synthesized multi-objective gain-scheduled noisy output-feedback (GSNOF) controller compensates for the gradually-degraded sensor performance due to sensor aging and maintains the desired performance given set of available control input constraints.

1.4 Chapter 6 Summary

This chapter presents the design a gain-scheduled controller, with guaranteed stability, that utilizes the sensor aging information to minimize the control effort subject to performance constraint on output covariance matrix (OCC) in the presence of gradual sensor performance degradation due to sensor aging. The OCC control problem is to minimize the control input covariance subject to the output covariance constraint(s). Without considering the output covariance matrix (OCC), it is possible to design a controller that utilizes minimum control effort to achieve the unsatisfactory output performance. On the other hand, with the OCC control design, the constraints on the regulated output are considered during the controller synthesis process. Therefore, the controller with the minimum control effort is obtained for a given set of constraints on the regulated output. The designed controller satisfies the required performance where the sensor performance degradation information is considered.

In summery, an overview of the main parts of this work are presented. Next chapter introduces the main notations and mathematical preliminaries were used in this work.

Chapter 2

Preliminaries

The aim of this chapter is to briefly introduce necessary notations and definitions associated with the modeling approach that is used in this dissertation. This chapter does not present any theoretical contributions but it is used to present notations, terminologies, and definitions that are used throughout this dissertation. Most of the terminologies used in this chapter can be found in [Rodrigues et al., 2018, White et al., 2013c, De Caigny et al., 2010].

2.1 Notation

Notations used in this dissertation are fairly standard. The real, positive real, discrete and positive discrete numbers are denoted by $\mathbb{R}$, $\mathbb{R}_{+}$, $\mathbb{Z}$ and $\mathbb{Z}_{+}$, respectively. The symbol $||\cdot||$ denotes the Euclidean norm unless it is stated otherwise. The symbol $\mathbb{E}[\cdot]$ denotes the exception operator. The symbol "$\mapsto$" denotes the mapping from one domain to another, while the symbol "$\rightarrow$" denotes the implying logic. "*co*" denotes the column of a matrix. The capital bold latter **J**, for example, denotes a matrix. The relation $\mathbf{J}< 0$ ($\mathbf{J} \leq 0$) means that the matrix **J** is negative (negative semi-) definite. $trace(\mathbf{A})$ denotes the trace of the matrix **A**, which represents the sum of diagonal elements of the matrix **A**. The zero matrix of size $n \times p$ is referred to as $\mathbf{0}_{n\times p}$. These subscripts will be omitted when the size of the corresponding matrix can be inferred from the context.

2.2 Definitions and Terminologies

Before going forward some definitions and terminologies need to be expressed.

Definition 1 *The expectation of a random variable [Shynk, 2012] is one measure of the location and center of its probability density function (pdf). It is the "expected value" in the sense that with repeated trials, the expectation would be observed on average, which is also called the mean. Let $\mathcal{R}$ be a random variable with a finite number of finite outcomes $r_1, \cdots, r_n$ occurring with probabilities $p_1, \cdots, p_n$, respectively. The expectation of $\mathcal{R}$ is defined as*

$$\mu_{\mathcal{R}} = \mathbb{E}(\mathcal{R}) = \sum_{i=1}^{n} r_i \, p_i \tag{2.1}$$

Since all probabilities add up to 1, the expected value is the weighted average, with p_i's being the weights. If all outcomes are equiprobable, then the weighted average turns into the simple average. This is intuitive: the expected value of a random variable is the average of all values it can take; thus the expected value is what one expects to happen on average.

Definition 2 *Variance of a random variable [Shynk, 2012] is a measure of spread for a distribution of a random variable that determines the degree to which the values of a random variable differ from the expected value. The variance of random variable $\mathcal{R}$ is*

$$\sigma_{\mathcal{R}}^2 = \mathbb{E}\left[\left(\mathcal{R} - \mu_{\mathcal{R}}\right)^2\right] \tag{2.2}$$

where $\mu_{\mathcal{R}}$ is the mean of the random variable $\mathcal{R}$. Note that, The standard deviation $\sigma_{\mathcal{R}}$ of random variable $\mathcal{R}$ is the positive square root of the variance $\sigma_{\mathcal{R}}^2$.

Definition 3 *The covariance can be viewed as an extension of the variance of one random variable. The covariance of random variables $\mathcal{R}$ and $\mathcal{Y}$ is*

$$Cov\big(\mathcal{R}\mathcal{Y}\big) = \mathbb{E}\big[\big(\mathcal{R} - \mu_{\mathcal{R}}\big)\big(\mathcal{Y} - \mu_{\mathcal{Y}}\big)\big] \tag{2.3}$$

Note that, when random variables $\mathcal{R}$ and $\mathcal{Y}$ are uncorrelated $Cov\big(\mathcal{R}\mathcal{Y}\big) = 0$.

Definition 4 *Unit-simplex [Nesterov, 2013]: a unit simplex is defined as follows*

$$\Lambda_r = \{\zeta \in \mathbb{R}^r : \sum_{i=1}^{r} \zeta_i = 1,\ \zeta_i \geq 0,\ i = 1, \cdots, r\}, \tag{2.4}$$

where the variable ζ_i varies in the unit-simplex Λ_r that have r vertices.

Definition 5 *Multi-simplex [Nesterov, 2013]: a multi-simplex Λ_M is defined as the Cartesian product of finite number of n simplex such as*

$$\Lambda_{M_1} \times \Lambda_{M_2} \times \cdots \times \Lambda_{M_n} = \prod_{j=1}^{n} \Lambda_{M_i} \triangleq \Lambda_M, \tag{2.5}$$

The dimension of the Λ_M is defined as the index $M = (M_1, M_2, \cdots, M_n)$ and for simplicity of notation, $\mathbb{R}^M$ denotes for the space $\mathbb{R}^{M_1+M_2+\cdots+M_n}$.

Remark 6 *Any variable ζ in the multi-simplex Λ_M can be decomposed in to $(\zeta_1, \zeta_2, \cdots, \zeta_n)$, in sequel, each ζ_i is part of the unit-simplex domain Λ_{M_i}, that can be decomposed into $(\zeta_{i1}, \zeta_{i2}, \cdots, \zeta_{iM_i})$, $\forall i = 1, \cdots, n$*

2.3 LPV Modeling Approaches

This section illustrates the general modeling approaches for the linear parameter varying (LPV) systems. That is, polytopic and affine parameterizations.

Definition 7 *Affine parameter-dependent : A matrix* $\mathbf{X}(\theta(k))$ *is considered as an affine parameter-dependent, if it can be represented such as*

$$\mathbf{X}(\theta(k)) = \mathbf{X}_0 + \sum_{i=1}^{n_x} \theta_i(k)\,\mathbf{X}_i, \tag{2.6}$$

where $\mathbf{X}_0 \in \mathbb{R}^{n\times n}$ *and* $\mathbf{X}_i \in \mathbb{R}^{n\times n}$ *are constant matrices; and* $\theta_i(k)$ *is the* i^{th} *element of the scheduling parameters vector* $\theta(k) \in \mathbb{R}^{n_x}$ *at time instant* k*.*

Definition 8 *Polytopic parameter-dependent : A matrix* $\mathbf{X}(\theta(k))$ *is considered as a polytopic parameter-dependent, if it can be represented such as*

$$\mathbf{X}(\alpha(k)) = \sum_{i=1}^{n_q} \alpha_i(k)\,\mathbf{X}_i, \tag{2.7}$$

where $\mathbf{X}_i \in \mathbb{R}^{n\times n}$ *are constant matrices; and* $\alpha_i(k)$ *is the* i^{th} *element of the scheduling parameters vector* $\alpha(k) \in \mathbb{R}^{n_q}$ *at time instant* k*.*

Chapter 3

Online Sensor Aging Detection using a Modified Adaptive Filter

3.1 Introduction

Due to the increasing complexity of modern engineering systems, reliability has become an increasingly important matter. This is particularly important for safety-critical applications, where a successful mission of protecting human life, property, and/or environment becomes a paramount goal. To minimize the possibility of unexpected failures, control systems need to have increased reliability. One way to improve the system reliability is to enhance its ability to fault detection [Ram and Davim, 2018]. There are many system failure sources; the most common and significant one is the sensor failure. To be specific, sensor performance degradation due to sensor aging. Sensor failure affects the system output performance directly and could lead to catastrophic consequences [Cai and Wu, 2010]. Indeed, a faulty sensor may cause system performance degradation, system shut down, and/or fatal accident; see Refs. [Zhang et al., 2017] and [Edwards et al., 2010] for faulty sensors in aircraft systems.

The recent fault detection literature focus on the effect of sensor fault to the system performance in many application fields, such as aircraft engines [Lu and Wu, 2009], altitude sensor fault of air-jet pitch and attach angle control system [Nguyen et al., 2017], wireless sensor networks [Alvergue et al., 2016], unmanned vehicles [Hajiyev et al., 2015], actuator faults [Tao et al., 2017], underwater vehicles [Liu et al., 2018], automobile engine mass air flow sensor [Tan et al., 2018], drones and satellite communication applications [Edwards et al., 2010], medical applications [Abdallah et al., 2018], wind power generation [Bagherieh and Nagamune, 2014], [Abdelmalek et al., 2018], Bayesian belief network [Mehranbod et al., 2005], small autonomous helicopters [Heredia et al., 2008], nuclear power plant applications [Mandal et al., 2017], security of cyber-physical systems [Cardenas et al., 2008], modern hybrid electric vehicles [Foo et al., 2013], and data management in time-series application [Sharma et al., 2010].

Different methods were used to model, detect, and identify sensor faults. The commonly used approaches in recent literature model the sensor fault as an additive or multiplicative term in the measurement equation, augment the fault terms with system states, and use filter (or observer) (e.g., generalized or unknown input observer, Kalman or H_∞ filter) to estimated the augmented states, a combination of system states and fault terms. Also, the sensor fault is modeled as an additive or multiplicative term and compared with the residual signal for a given threshold to detect the faulty sensor. The virtual sensor technique was used to mask the faulty sensor measurement(s) and keep the system within its nominal performance [Rotondo et al., 2014].

In contrast to recent literature, model the sensor fault due to aging as stationary augmented state, this work assumes that the sensor performance shift (i.e., sensor performance gradual-degradation and/or fault due to sensor aging) can be characterized by its measurement noise covariance variation, comparing with the most of the noise statistics identification literature for the case with time-invariant sensor noise covariance only. That is, the gradual-degradation of sensor performance due to sensor aging is characterized by the slowly-varying sensor measurement noise covariance. To match with the goal of online monitoring the sensor performance degradation due to sensor aging, an algorithm is developed to detect sensor performance degradation due to sensor aging that is characterized by the slowly-varying sensor noise covariance matrix, that is, online estimating slowly-varying sensor measurement noise covariance. Indeed, filtering theory can be used to monitor the sensor performance and detect its performance degradation. The resulting filter to deal with this matter is known as an adaptive filter that adjusts its parameters according to the gap between the predicted estimates and the current measurements, for example, adaptive Kalman filter that has been investigated by many kinds of literature. To be specific, adaptive Kalman filter estimates the states with a noisy measurement. For example, in a networked control system, where the communication channels between the sensors and the remote estimator are vulnerable to false data injection attacks, as presented in Ref. [Hu et al., 2018]. An optimal model-free control policy for tracking problem of non-affine nonlinear multi-input-multi-output (MIMO) systems was proposed in Ref. [Safaei and Mahyuddin, 2017], where all the states in the system are measured using a set of sensors, with white noise. In addition, adaptive Kalman filtering algorithm is designed for the road environment to track radar targets and improve the accuracy of target tracking, where the radar has more noise interference due to the changeable road environment and complex background [Zhai et al., 2018]. However, the unknown noise statistics can be estimated online from the observed data. For example, a proposed reference recursive recipe for estimating initial filter state, process noise covariance and the unknown parameters is demonstrated in Ref. [Ananthasayanam, 2018]. In Ref. [Zheng et al., 2018] a robust adaptive Unscented Kalman filter (RAUKF) to improve the accuracy and robustness of state estimation with uncertain noise covariance. Note that, the last two references deal with the case of time-invariant noise covariance. Considering that the gradual sensor performance degradation due to sensor aging can be characterized by the slowly-varying sensor noise covariance. This inspires this work to investigate the estimation of the time-varying sensor noise covariance, or equivalently, covariance matrices related to the state and measurement models. Consequently, the proposed algorithm invests adaptive filtering theory with the covariance-matching technique for such goal. The basic idea behind using the covariance-matching technique [Meng et al., 2016] is to make the innovation sequences consistent with their theoretical covariance. The covariance-matching method has been shown to be one of the most promising techniques for practical applications due to its simplicity and online applicability [Meng et al., 2016]. For example, Ref. [Zhang et al., 2018] proposes an adaptive filter for joint polarization tracking and channel equalization using cascaded covariance-matching. Ref. [Chen et al., 2017] proposes a target tracking algorithm using time difference of arrival and frequency difference of arrival measurements for a mobile target in a distributed sensor network, where the prior noise covariances are uncertain. A sensitivity-based adaptive square-root unscented Kalman filter (SRUKF) with the recursive prediction-error method was used to estimate system states, parameters, and covariances online is presented in Ref. [Riva et al., 2018].

In this chapter the primary goal is to monitor sensor performance and detect sensor fault, that is, gradual sensor performance degradation due to the sensor aging characterized by senor slowly-varying measurement noise covariance using the proposed algorithm.

In contrast to fault detection literature, this work characterizes the gradual sensor performance degradation by slowly-varying sensor noise covariance. In addition, up-to authors knowledge, most of the noise statistics identification literature considering the case with time-invariant sensor noise covariance only, and this work extends to the case with time-varying sensor noise covariance due to sensor aging. That is, the gradual sensor performance degradation due to sensor aging (i.e., sensor fault due to sensor aging) is characterized by the slowly-varying sensor measurement noise covariance. Note that, there may be more sophisticated methods to estimate the sensor noise statistics, such as Bayesian analysis, which need more time consumption and computation complexity (i.e., more computation load) and in consequence, they are not convenient for online application. Moreover, the proposed algorithm is suitable for online applications due to its iterative manner, in contrast to most popular filtering recursive manner algorithms, is less sensitive to initial conditions (i.e., less sensitive to initial state and its initial estimation error covariance matrix). In other words, the proposed algorithm could be used as an online sensor health-monitoring, and fault detection technique, which helps to avoid system performance degradation, system shut down, and/or fatal accident; for more details see [Madhag and Zhu, 2017] or [Madhag and Zhu, b].

As a contribution of this chapter, an algorithm is proposed to monitor and detect gradual sensor performance degradation due to sensor aging that is characterized by slowly-varying sensor noise covariance. Additionally, this chapter proposes the incorporation of the innovation vector sequence quality information with the weighted measurements, used by the proposed filter, for estimating the slowly-varying sensor noise covariance. Moreover, the proposed algorithm is suitable for online applications due to its iterative manner; use of weighted information about the innovation vector sequence quality with reduced computational load and the exponentially weighted estimation window improves the estimation accuracy. Consequently, the proposed algorithm could also be used as an online fault identification technique, to decide when the sensor fails and its measurements are useless. That is particularly important for safety-critical applications, where a successful mission of protecting human life, property, and/or environment becomes a paramount goal.

The next Sub-section provides a general overview of the system model and describes the proposed approach for estimating sensor noise covariance using adaptive Kalman filter based on a weighted innovation vector and whiteness test along with covariance-matching. In Sub-section 3.4 a simulation example is used to express the capability of the proposed approach and the bounds of the estimation window. Conclusions are drawn in Sub-section 3.5.

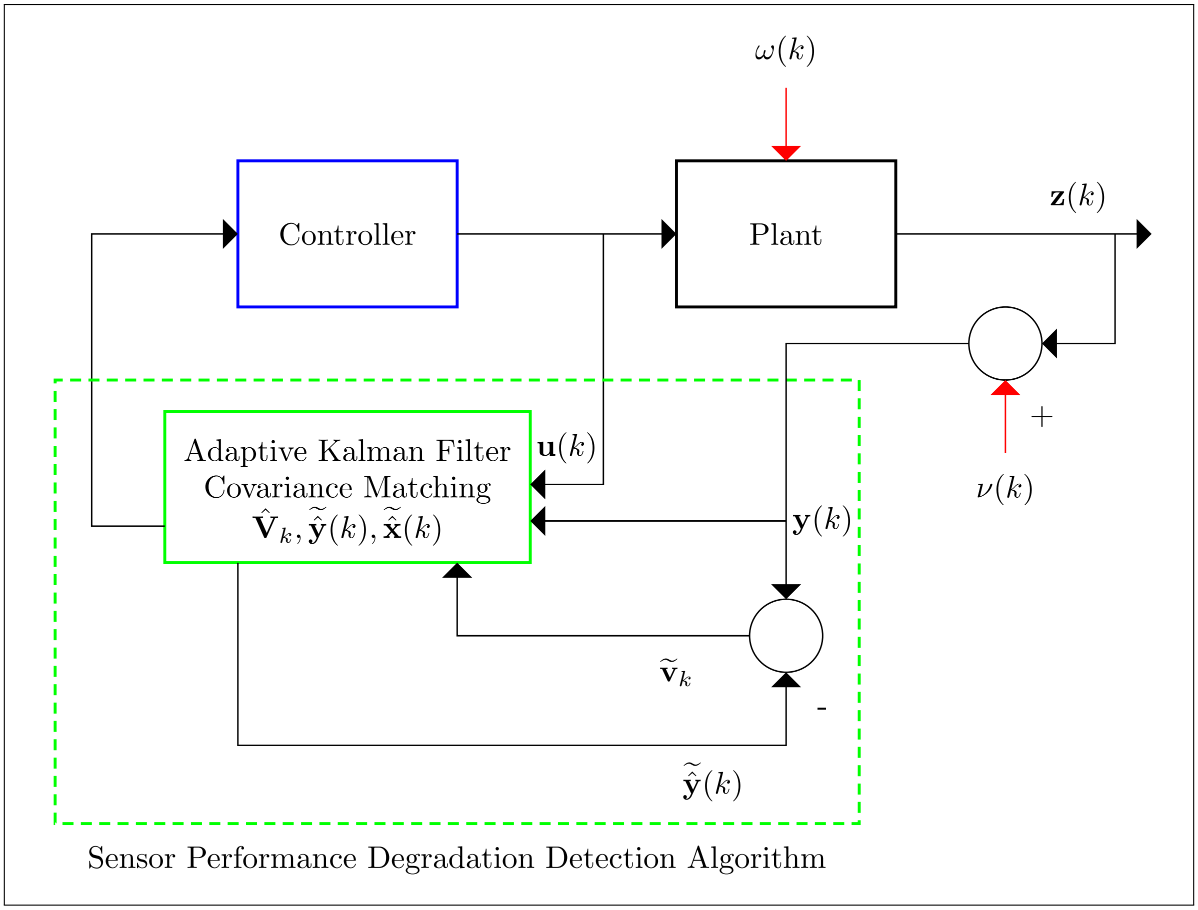


Figure 3.1: Stochastic closed-loop system, where x, y and u are the state, the measurement and the input vectors, respectively; $\tilde{\hat{x}}$ and $\tilde{\hat{y}}$ are the estimated state and the estimated measurement vectors, respectively; $\tilde{\mathbf{v}}$ is the innovation vector ; ω and ν are the process and measurement sensor noises, respectively; The superscripts tilde and head denote the sub-optimal and the associated estimated value, respectively. The dotted-line represents the controller part that is known.

3.2 System Model

To describe the proposed algorithm clearly, the dynamic system model considered and the associated assumptions are reviewed first. The target discrete-time stochastic closed-loop system is shown in Fig. 3.1, where the dotted-line represents the controller part that is assumed known at this chapter.

The plant model is described in the following state-space representation

$$\begin{cases} \mathrm{x}(k+1) &= \mathbf{A}(k)\,\mathrm{x}(k) + \mathbf{B}_u(k)\,\mathrm{u}(k) + \mathbf{B}_\omega(k)\,\omega(k)\,, \\ \mathrm{z}(k) &= \mathbf{C}_z(k)\,\mathrm{x}(k) + \mathbf{D}_z(k)\,\mathrm{u}(k), \\ \mathrm{y}(k) &= \mathbf{C}_y(k)\,\mathrm{x}(k) + \ \nu(k)\,, \end{cases} \tag{3.1}$$

where subscript k is the sample time index; $\mathrm{x}(k) \in \mathbb{R}^n$ is the state vector; $\mathrm{z}(k) \in \mathbb{R}^q$ is the controlled output vector; $\mathrm{y}(k) \in \mathbb{R}^p$ is the measured output vector; $\mathrm{u}(k) \in \mathbb{R}^m$ is the input vector; $\mathbf{A}(k) \in \mathbb{R}^{n\times n}$ is the system matrix; $\mathbf{B}_u(k) \in \mathbb{R}^{n\times m}$ is the input matrix; $\mathbf{C}_z(k) \in \mathbb{R}^{q\times n}$; $\mathbf{D}_z(k) \in \mathbb{R}^{q\times m}$ and $\mathbf{C}_y(k) \in \mathbb{R}^{p\times n}$ are the measurement matrices; and $\mathbf{B}_\omega(k)$ is a diagonal matrix with proper dimension. Indeed, there are two interesting interpretation of the terms $\omega(k)$ and $\nu(k)$ in system (3.1). Deterministic interpretation: the exogenous input vectors $\omega(k)$ and $\nu(k)$ are assumed to be an unknown disturbance that belongs to a bounded $\mathcal{L}_2$ set. Stochastic interpretation: they are assumed to be uncorrelated Gaussian white noises with their means and covariances are defined by

$$\begin{cases} \mathbb{E}\left[\omega(k)\right] = 0, \;\; \mathbb{E}\left[\omega(k)\omega(j)^T\right] = \mathbf{W}_k\,\delta(t_k - t_j) \\ \mathbb{E}\left[\nu(k)\right] = 0, \;\; \mathbb{E}\left[\nu(k)\nu(j)^T\right] = \mathbf{V}_k\,\delta(t_k - t_j), \\ \mathbb{E}\left[\omega(k)\nu(j)^T\right] = 0, \;\; \mathbb{E}\left[\mathrm{x}(k)\nu(j)^T\right] = 0, \quad \forall\, k, j \in \mathbb{R}, \end{cases} \tag{3.2}$$

where $\mathbb{E}\left[\cdot\right]$ denotes the expectation operator; δ is a Kronecker Delta function (i.e., $\delta(0) = 1$ and $\delta(k) = 0\ \forall\, k \neq 0$ otherwise); and $\mathbf{W}_k$ and $\mathbf{V}_k$ are time-varying process and measurement noise covariance matrices, respectively. This work considers only the stochastic interpretation since our goal to detect the sensor performance degradation or fault. Note that the Kronecker Delta function is used for discrete-time systems and for continuous-time case a Dirac Delta Function should be used.

The process used to simulate and model the sensor performance degradation due to aging is carried out under the following assumptions. The initial state vector x(0) in system (3.1) is assumed to have a normal distribution with zero mean ($\mathbb{E}[\mathrm{x}(0)] = 0$) and covariance matrix $\mathbf{P}_0 = \mathbb{E}(\mathrm{x}(0)\mathrm{x}^T(0)) > 0$. The pairs $\left(\mathbf{A}(k), \mathbf{C}_y(k)\right)$ and $(\mathbf{A}(k), \mathbf{B}(k))$ are assumed to be uniformly observable and controllable, respectively. The process noise covariance matrix, $\mathbf{W}_k$, is assumed to be positive semi-definite and known at each time instant. The measurement noise covariance, $\mathbf{V}_k$, is assumed to be positive definite to be estimated. In the numerical calculations, the negative eigenvalues of estimated sensor noise covariance matrix are replaced by small positive values. Moreover, all sensors measurement noises of all channels are assumed to be independent (i.e., measurement noise for each channel is independent on these for the rest of measurements), that is, $\mathbf{V}_k$ is a diagonal matrix.

Remark 9 *Kalman filter is very sensitive to the positive definiteness of measurement noise covariance matrix, and non-definiteness could affect estimation convergence and lead to divergence. In the numerical calculations, the negative eigenvalues of the estimated measurement noise covariance matrix are replaced by small positive values.*

Remark 10 *There are three time scales for system* (3.1). *Time scale 1, S_1, is represented by the sensor noise covariance that has the slowest rate of variations; Time scale 2, S_2, is represented by the time-varying coefficients of system matrices, $\mathbf{A}(k), \mathbf{B}_u(k), \mathbf{C}_z(k), \mathbf{C}_y(k)$, and $\mathbf{B}_\omega(k)$, that have a slowly time-varying coefficients with much faster rate of variations than that of sensor noise covariance; finally, Time scale 3, S_3, is represented by the system dynamics, that has the fastest rate of variations and is much fast than the other two time scales in the system.*

Remark 11 *The sensor noise covariance variation is used to simulate practical engineering problems. For example, the sensor noise covariance varying linearly, exponentially, or linearly with* *sinusoid fluctuation represents an aging sensor that has measurement noise (or error) increases linearly, exponentially, or linearly with sinusoid fluctuation.*

The aforementioned assumptions and remarks are required to be satisfied to proceed with the sensor performance degradation estimation (i.e., sensor noise covariance estimation). The system controllability and observability assumptions are essential to sensor noise covariance estimation. If the system has unstable, undetectable, or/and unstabilizable modes, the sensor noise covariance estimation will deviate far away from the actual one. The assumptions to the process and measurement noise covariances are necessary for estimating sensor noise covariance online. The proposed algorithm through this chapter requires the three time scales to have an ascending order of rate of variations with large separations (at least one order difference). That is, sensor noise covariance has the slowest rate of variations, system matrix coefficients have a slow rate of variations, and system dynamics has the fastest rate of variations among three.

3.3 Adaptive Sub-Optimal Filter with Covariance Matching Technique

In this chapter the primary goal is to online monitor sensor performance degradation, and detect sensor failure, that is, online estimating the slowly-varying sensor measurement noise covariance due to the sensor aging. To achieve such goal, we invest the modified adaptive Kalman filter with covariance-matching technique for a system with measurement noise covariance varying slowly. The gradual-degradation of sensor performance due to sensor aging is characterized by its slowly-varying measurement noise covariance. In particular, it usually cannot obtain the exact statistical properties of the sensor noise because it is impossible to isolate the noise from the measurement signals accurately [Kay, 2013]. Also, in most cases, the mechanism of the noise is not completely understandable [Kay, 2013]. References [Simon, 2006], [Grewal and Andrews, 2001] show that using the noise covariance different from the actual one in the Kalman filter affects the accuracy of the state estimation since the Kalman filter is not optimal anymore, and Kalman filter (3.3) under such conditions is a sub-optimal one; see Refs. [Grewal and Andrews, 2001] and [Simon, 2006] for more details. The adaptive scheme with Kalman filter is used to reduce or bound errors by modifying or adapting the Kalman filter to the actual measurements. That is, the adaptive Kalman filter adjusts its estimation gains based on the gap between the predicted estimates and the current measurements. The adaptive Kalman filter [Grewal and Andrews, 2001], [Simon, 2006] is given below

$$
\begin{cases}
\widetilde{\hat{\mathrm{x}}}(k|k-1) = \mathbf{A}(k)\,\widetilde{\hat{\mathrm{x}}}(k-1|k-1) + \mathbf{B}_u(k)\,\mathrm{u}(k), \\
\widetilde{\hat{\mathbf{P}}}_{k|k-1} = \mathbf{A}_{CL}(k)\,\widetilde{\hat{\mathbf{P}}}_{k-1|k-1}\,\mathbf{A}_{CL}^T(k) + \mathbf{B}_\omega(k)\,\mathbf{W}_{k-1}\,\mathbf{B}_\omega^T(k)\,,
\end{cases}
$$
$$
\begin{cases}
\tilde{\mathbf{v}}_k = \mathrm{y}(k) - \mathbf{C}_y(k)\,\widetilde{\hat{\mathrm{x}}}(k|k-1), \\
\widetilde{\mathbf{K}}_k = \widetilde{\hat{\mathbf{P}}}_{k|k-1}\,\mathbf{C}_y^T(k)\left[\mathbf{C}_y(k)\,\widetilde{\hat{\mathbf{P}}}_{k|k-1}\,\mathbf{C}_y^T(k) + \hat{\mathbf{V}}_{n,k}\right]^{-1},
\end{cases}
\tag{3.3}
$$
$$
\begin{cases}
\widetilde{\hat{\mathrm{x}}}(k|k) = \widetilde{\hat{\mathrm{x}}}(k|k-1) + \widetilde{\mathbf{K}}_k\,\tilde{\mathbf{v}}_k, \\
\widetilde{\hat{\mathbf{P}}}_{k|k} = \left(\mathbf{I} - \widetilde{\mathbf{K}}_k\,\mathbf{C}_y(k)\right)\widetilde{\hat{\mathbf{P}}}_{k|k-1}\left(\mathbf{I} - \widetilde{\mathbf{K}}_k\mathbf{C}_y(k)\right)^T + \widetilde{\mathbf{K}}_k\hat{\mathbf{V}}_{n,k}\widetilde{\mathbf{K}}_k^T,
\end{cases}
$$

where $\widetilde{\hat{\mathrm{x}}}$ and $\widetilde{\hat{\mathbf{P}}}$ are the estimated state and its estimation error covariance, respectively, using the sub-optimal Kalman filter, or in other words, Kalman filter with estimated noise covariance; $\mathbf{A}_{CL}(k)$ is the closed-loop state matrix; $\tilde{\mathbf{v}}_k$ is the innovation vector; $\widetilde{\mathbf{K}}_k$ is the sub-optimal Kalman gain; and $\hat{\mathbf{V}}_{n,k}$ is the estimated noise covariance matrix. The superscripts tilde and head denote the sub-optimal and the associated estimated value, respectively and other matrices are defined below

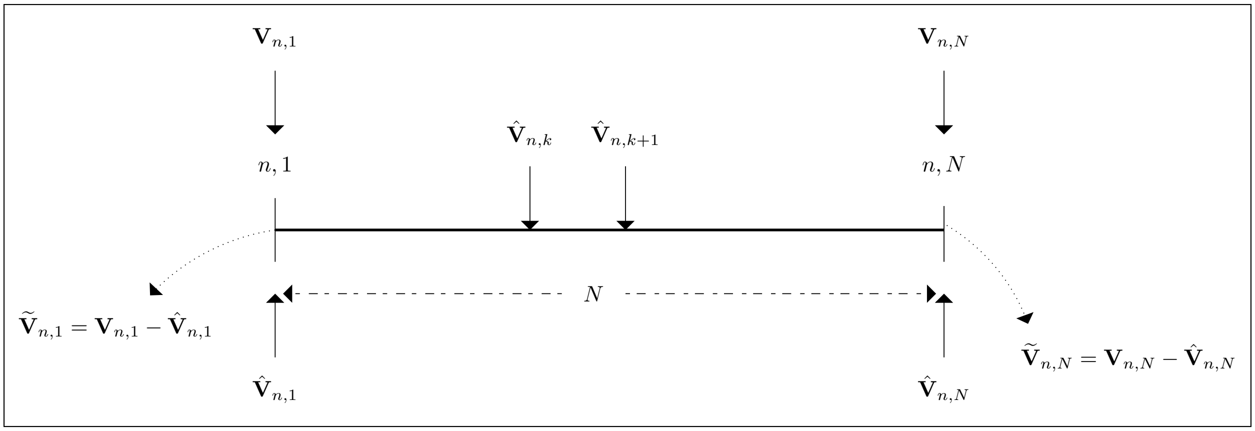


Figure 3.2: Estimation window specifications. The subscripts n and k denote the estimation window number and time within the estimation window, respectively. The length of the estimation window is denoted by N. The estimated noise covariance matrix at estimation window n and sample time k is denoted by $\hat{\mathbf{V}}_{n,k}$. The first and last time instants of the estimation window are denoted by "n, 1" and "n, N", respectively. The associated estimated noise covariances are denoted by $\hat{\mathbf{V}}_{n,1}$ and $\hat{\mathbf{V}}_{n,N}$, respectively, while any two consecutive values within the window are denoted by $\hat{\mathbf{V}}_{n,k}$ and $\hat{\mathbf{V}}_{n,k+1}$, respectively. $\mathbf{V}_{n,1}$ and $\mathbf{V}_{n,N}$ denote the actual noise covariances at first and last time instant of the estimation window.

system (3.1). To match with the goal of online monitoring sensor performance and detecting the gradual sensor performance degradation due to sensor aging (i.e., sensor fault due to sensor aging) characterized by the slowly-varying sensor noise covariance matrix, it is assumed that the system (3.1) is stable. In addition, equation (3.3) could be different from traditional ones since the Joseph formula is used to update the state estimation error covariance matrix (i.e., last line of group equations(3.3)), that is, it ensures fast convergence and upper bounded estimation error, see Ref. [Simon, 2006] for more details.

In this work notations that express the estimation window number and time indices are used. There is a fixed length time interval called estimation window sliding over the infinite time horizon. The length of the estimation window is denoted by N. Both subscripts n and k denote the estimation window number and sample time indices, respectively. The current estimation window is represented by n, the previous estimation window by $n-1$, and the next by $n+1$. The estimated noise covariance matrix at estimation window n and sample time k is denoted by $\hat{\mathbf{V}}_{n,k}$. The first and last time instants of the current estimation window are denoted by "$n, 1$" and "n, N", respectively. The associated estimated noise covariance matrices are denoted by $\hat{\mathbf{V}}_{n,1}$ and $\hat{\mathbf{V}}_{n,N}$, respectively, while any two consecutive estimated noise covariance matrices within the window are denoted by $\hat{\mathbf{V}}_{n,k}$ and $\hat{\mathbf{V}}_{n,k+1}$, respectively, the actual noise covariances use the same notation but without superscript head; see Fig. 3.2 for more details.

Adaptive Kalman Filtering (AKF) with covariance-matching technique considered in this work to online estimate the unknown sensor noise covariance matrix. Note that covariance-matching technique is highlighted in many publications such as [Meng et al., 2016] and others. The basic idea behind the covariance-matching technique is to make the innovation vector sequence (defined in equation (3.3)) consistent with their theoretical covariance matrix. Remember that the goal in this chapter is to online detect gradual sensor performance degradation due to sensor aging by estimating the slowly-varying sensor noise covariance from the set of measurements based on the proposed algorithm based on the information about the quality of the weighted innovation vector sequence. It is well known that the innovation vector sequence is a reliable indicator for the filtering performance [Chui and Chen, 2008], [Simon, 2006]. Note that the innovation vector is described as the difference between the actual measurements and its predicted values [Grewal and Andrews, 2001], [Simon, 2006] and it is given by

$$\tilde{\mathbf{v}}_k = \mathrm{y}(k) - \mathbf{C}_y(k)\,\widetilde{\hat{\mathrm{x}}}(k|k-1),$$

where $\widetilde{\hat{\mathrm{x}}}(k|k-1)$ is the estimated state vector at time k based on measurements at time $k-1$ using filter (3.3); $\mathrm{y}(k)$ is the measurement vector at time k; $\mathbf{C}_y(k)$ is the measurements matrix; $\tilde{\mathbf{v}}_k$ is the innovation vector, and the superscripts tilde and head denote the sub-optimal and estimated value, respectively.

Remark 12 *The innovation vector is said to be white if it is uncorrelated with zero mean.*

Two criteria are used to test whiteness of the innovation vector and they are bias and whiteness tests [Kay, 2013], [Bar-Shalom and Kirubarajan, 2004]. The sample mean for the collection of innovation vectors (or the innovation vector estimate)[Kay, 2013], [Bar-Shalom and Kirubarajan, 2004] is given by

$$\bar{\tilde{\mathbf{v}}}_N = \frac{1}{N}\sum_{k=1}^{N}\tilde{\mathbf{v}}_k \ . \tag{3.4}$$

where N is the number of time samples. The estimated auto-covariance matrix of the innovation vector [Kay, 2013], [Bar-Shalom and Kirubarajan, 2004] is given by

$$\mathrm{Cov}(\tau) = \frac{1}{N}\sum_{k=1}^{N-\tau}(\tilde{\mathbf{v}}_{k+\tau} - \bar{\tilde{\mathbf{v}}}_N)(\tilde{\mathbf{v}}_k - \bar{\tilde{\mathbf{v}}}_N)^T. \tag{3.5}$$

Note that use the divisor N instead of $N-\tau$ ensures that the auto-covariance matrix is nonnegative definite [Brockwell and Davis, 2016]. The bias test is used to check statistically whether the sample mean of the collection of the innovation vectors is zero or not. The whiteness test is to check statistically whether the estimated auto-covariance matrix of the innovation vector is within the confidence interval or not. The 95% confidence interval estimate of the auto-covariance matrix [Kay, 2013] is given by

$$\ell(\tau) = \left[\mathrm{Cov}(\tau) - \frac{1.96\times\mathrm{Cov}(0)}{\sqrt{N}} \ , \ \ \mathrm{Cov}(\tau) + \frac{1.96\times\mathrm{Cov}(0)}{\sqrt{N}}\right] \ . \tag{3.6}$$

Note that the auto-covariance matrix, $\mathrm{Cov}(\tau)$, with zero lag ($\tau = 0$) is the covariance matrix denoted by $\mathbf{P}_\mathrm{v}$ (i.e., $\mathrm{Cov}(0) = \mathbf{P}_\mathrm{v}$).

Remark 13 *The innovation vector sequence, used to estimate the sensor measurement noise covariance, has to pass both bias and whiteness tests.*

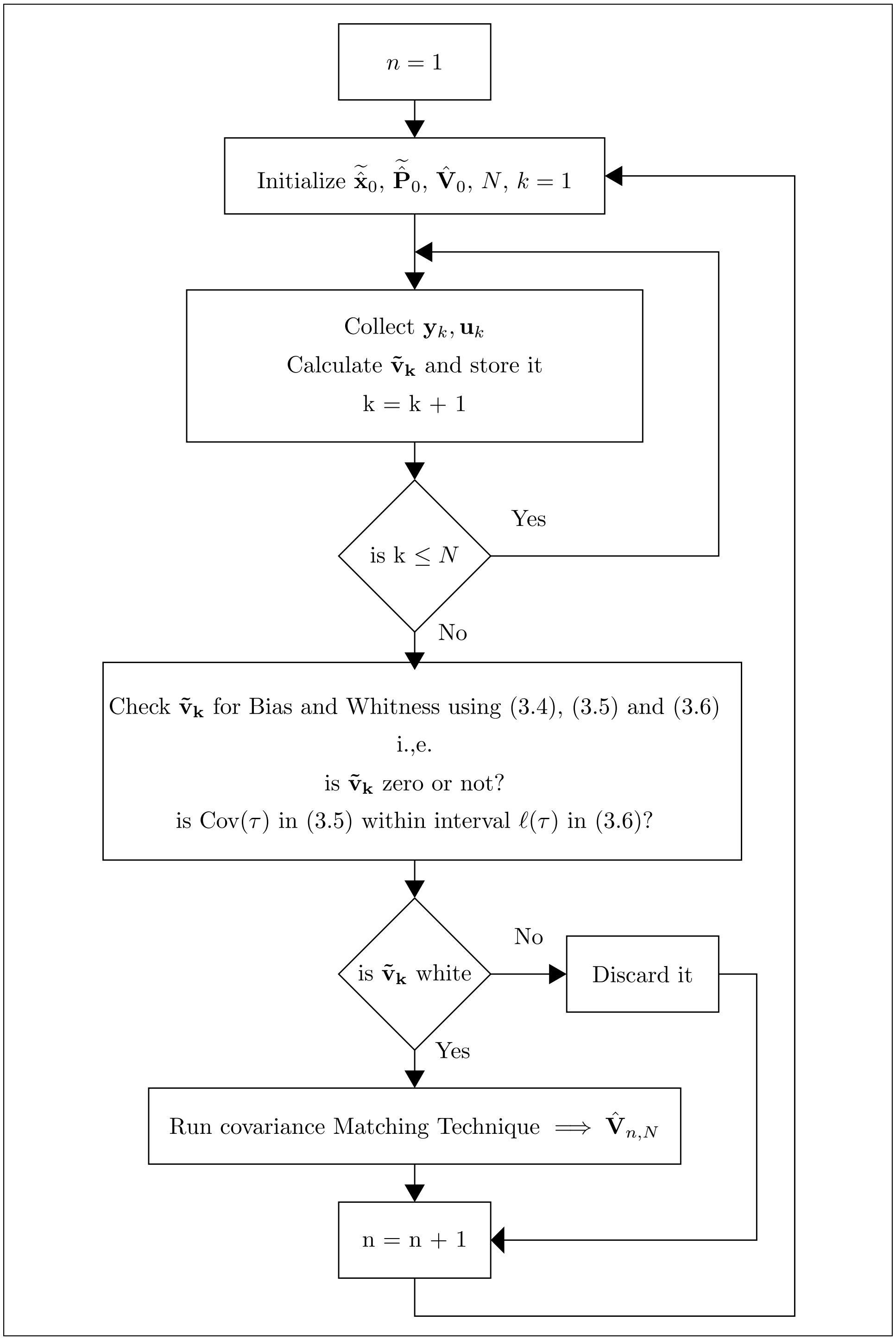


Figure 3.3: Flow chart of the slowly-varying sensor noise covariance estimation algorithm. n is the estimation window index; N is the estimation window size; $\widetilde{\hat{x}}(0)$ and $\widetilde{\hat{\mathbf{P}}}_0 = \mathbb{E}(\widetilde{\hat{x}}(0)\widetilde{\hat{x}}(0)^T)$ are the initial estimated state and its estimation error covariance; k is the sample time index within the estimation window; u(k), y(k) and $\tilde{\mathbf{v}}_k$ are input, measurement and innovation, respectively; $\hat{\mathbf{V}}_o$ is the initial noise covariance estimate; $\hat{\mathbf{V}}_{n,N}$ is the estimated sensor noise covariance matrix by the end of estimation window "n".

The proposed algorithm uses the information about whiteness and the weighted innovations vectors (which will be explained later) for the adaptive estimation of the sensor noise covariance. Fig. 3.3 shows the proposed algorithm, where n is the estimation window index; N is the estimation window size; $\widetilde{\hat{\mathrm{x}}}_0$ and $\widetilde{\hat{\mathbf{P}}}_0$ are the initial estimated state and its estimation error covariance; k is the sample time index within the estimation window; $\mathrm{u}(k)$, $\mathrm{y}(k)$, and $\tilde{\mathbf{v}}$ are input, measurement, and innovation, respectively; $\hat{\mathbf{V}}_o$ is the initial noise covariance matrix estimate; and $\hat{\mathbf{V}}_{n,N}$ is the estimated sensor noise covariance matrix by the end of estimation window "n". A particularly attractive feature of combining the adaptive Kalman filter with the quality of the innovation vector information is that the filter is acting like a detection algorithm identifying not only changes in the system properties but also covariance in the disturbance statistics. Note that $\tilde{\mathbf{v}}_k$ is the innovation vector while $\nu(k)$ is the sensor measurement noise. Consider system (3.1) and filter (3.3), the exact value of the sensor measurement noise, $\nu(k)$, is unknown, an intuitive approximation of $\nu(k)$ is the innovation vector [Grewal and Andrews, 2001], [Simon, 2006], that is given by

$$\tilde{\mathbf{v}}_k = \mathrm{y}(k) - \mathbf{C}_y(k)\,\widetilde{\hat{\mathrm{x}}}(k), \tag{3.7}$$

where $\widetilde{\hat{\mathrm{x}}}(k)$ is $\widetilde{\hat{\mathrm{x}}}(k|k-1)$ defined in (3.3). Substitute $\mathrm{y}(k) = \mathbf{C}_y(k)\,\mathrm{x}(k) + \nu(k)$ in Eqn. (3.7), multiply by its transpose and take expectation with considering set of Eqns. (3.2) (i.e., the sate estimation error and measurement noise are uncorrelated), the calculated covariance matrix of the innovation estimate is given in by

$$\begin{aligned} \mathbb{E}\left[\tilde{\mathbf{v}}_k\tilde{\mathbf{v}}_k^T\right] &= \mathbb{E}\left[\mathbf{C}_y(k)\,(\mathrm{x}(k) - \widetilde{\hat{\mathrm{x}}}(k|k-1))\,(\mathrm{x}(k) - \widetilde{\hat{\mathrm{x}}}(k|k-1))^T\,\mathbf{C}_y^T(k)\right] + \mathbb{E}\left[\nu(k)\nu(k)^T\right], \\ &= \mathbf{C}_y(k)\widetilde{\hat{\mathbf{P}}}_{k|k-1}\mathbf{C}_y^T(k) + \mathbf{V}_k \end{aligned} \tag{3.8}$$

The sample mean of the collection of the innovation vectors (or innovation vector estimate)[Kay, 2013], [Bar-Shalom and Kirubarajan, 2004] is given by

$$\bar{\tilde{\mathbf{v}}}_N = \frac{1}{N}\sum_{k=1}^{N} \tilde{\mathbf{v}}_k, \tag{3.9}$$

where N is the number of time samples. The estimated covariance matrix of the innovation vector estimate is given in by

$$\mathbf{P}_{\mathrm{v}} = \frac{1}{N-1}\sum_{k=1}^{N} (\tilde{\mathbf{v}}_k - \bar{\tilde{\mathbf{v}}}_N)(\tilde{\mathbf{v}}_k - \bar{\tilde{\mathbf{v}}}_N)^T.$$

Note that the divisor is $N-1$ instead of N to ensure unbiased estimate of the covariance matrix $\mathbf{P}_{\mathrm{v}}$ [Kay, 2013]. Then, the estimate of the sensor measurement noise covariance at estimation window n with sample time $k = N$ is given by difference between the estimated and the calculated covariance matrix of the innovation vector estimate as follows

$$\hat{\tilde{\mathbf{V}}}_{n,N} = \frac{1}{N-1}\sum_{k=1}^{N}\Big[(\tilde{\mathbf{v}}_k - \bar{\tilde{\mathbf{v}}}_N)(\tilde{\mathbf{v}}_k - \bar{\tilde{\mathbf{v}}}_N)^T - \left(\frac{N-1}{N}\right)\mathbf{C}_y(k)\widetilde{\hat{\mathbf{P}}}_{k|k-1}\mathbf{C}_y^T(k)\Big], \tag{3.10}$$

where $\tilde{\mathbf{v}}_k$ is the innovation vector; $\bar{\tilde{\mathbf{v}}}_N$ innovation vector estimate, and $\widetilde{\hat{\mathbf{P}}}_{k|k-1}$ state estimation error covariance. Equation (3.10) is used to online estimate the sensor measurement noise covariance. Indeed, to obtain a feasible estimation based on the available measurements (minimal available information) for reducing estimation error and computational load, the last estimation is then combined with a weighted innovation vector. The weight coefficient φ is used as a "forgetting factor" for the past innovation vector data, and it is adaptively adjusted to improve the estimation accuracy as shown later. The weighted innovation vector estimate in equation (3.9) can be rewritten as follows

$$\bar{\mathbf{v}}_N = \frac{1}{N}\left[\sum_{k=1}^{N}\varphi^{N-k}\,\tilde{\mathbf{v}}_k\right], \tag{3.11}$$

where N is number of time samples; $\varphi \in (\epsilon, 1]$ is the forgetting factor for the innovation vector data in the past and $0 < \epsilon < 1$; $\bar{\mathbf{v}}_N$ is the innovation vector estimate defined in equation (3.9) but with the forgetting factor. Then, equation (3.10) can be written as

$$\hat{\mathbf{V}}_{n,N} = \frac{1}{N-1}\sum_{k=1}^{N}\left[(\varphi^{N-k}\tilde{\mathbf{v}}_k - \bar{\mathbf{v}}_N)(\varphi^{N-k}\tilde{\mathbf{v}} - \bar{\mathbf{v}}_N)^T - \left(\frac{N-1}{N}\right)\mathbf{C}_y(k)\widetilde{\hat{\mathbf{P}}}_{k|k-1}\mathbf{C}_y^T(k)\right], \tag{3.12}$$

where $\hat{\mathbf{V}}_{n,N}$ is the estimate of the sensor measurement noise covariance defined in equation (3.10) but with the forgetting factor.

In the proposed algorithm, the filter performance varies as the forgetting factor φ changes. That is, the value of the forgetting factor φ is important to the whole filtering process. Note that, the role of the forgetting factor φ at each estimation window is to put more weight on recent innovation data and less to older ones within that estimation window. Then, it is different from traditional forgetting factor or exponential weighting techniques that consider the weight over entire time horizon. The value of the forgetting factor φ is chosen to ensure the convergence of the steady-state mean square error. Usually φ is a number less than 1. Specifically, it is commonly selected to be $0.95 \leq \varphi \leq 0.99$ [Raol and Gopal, 2012],[Chu, 2015]. Consider the innovation vector given by equation (3.7) representing the output estimation error based on the predicted one, and square sum over the estimation window, as follows

$$\begin{aligned}\varepsilon_k &= \frac{1}{N}\Big(\sum_{k=1}^{N}\varphi^{N-k}\,(\tilde{\mathbf{v}}_k)^2\Big),\\ &= \frac{1}{N}\Big(\varphi^{N-1}\,(\tilde{\mathbf{v}}_1)^2 + \varphi^{N-2}\,(\tilde{\mathbf{v}}_2)^2 + \cdots + \varphi\,(\tilde{\mathbf{v}}_{N-1})^2 + (\tilde{\mathbf{v}}_N)^2\Big),\end{aligned}$$

where ε_k is the sum of mean-square-error. It is clear that as the forgetting factor should be less than 1 to have a monotonically decreasing sum of mean-square-error. If φ is small, the estimation tracking is strong [Kadlec et al., 2011]. On the other hand, the effect of early error and innovation are forgotten and since Kalman filter depends on innovation data to update the estimate, the current update step ignores new information. Consequently, the resulting estimate deviates from the actual value, leading to diverged filter estimation error. Indeed, when the filter starts at each estimation window, the estimation quality is low and it takes more iteration (information) to improve the estimation. As a result, the estimation should be weighting less at the beginning and more currently. In other words, when φ is close to one, the Kalman filter converges to a steady-state error in a slow manner, yielding a small error, and when is close to zero, the algorithm converges to a steady-state error with a relatively fast manner, yielding a large error. To overcome this shortcoming, it is usually required that $0.95 \leq \varphi \leq 0.99$. In this work, we consider forgetting factor values as follows [Kaufman et al., 2012], [Narendra, 2012]

$$\varphi = \frac{N-1}{N},$$

where, N is the estimation window size. Note that, by considering the forgetting factor for the innovation data vector, the estimation window is exponentially weighted, providing improved estimation accuracy. Now, we are in the position to state the Lemma of the proposed algorithm.

Lemma 14 *Consider system* (3.1) *and filter* (3.3) *with aforementioned assumptions and follow the algorithm defined in Fig. 3.3, the estimation of the sensor measurement noise covariance is given by equation* (3.12).

The last Lemma states the proposed estimation algorithm for the slowly time-varying sensor noise covariance using the adaptive Kalman filter based on the covariance-matching technique. Next, a simulation example is used to validate the proposed approach for monitoring sensor performance and estimating gradual sensor performance degradation due to sensor aging, that is, estimating the slowly-varying sensor noise covariance matrix due to sensor aging, of a given system.

3.4 Simulation Results and Discussion

The objective of conducting simulation study using a numerical example in this sub-section is to demonstrate the ability of the proposed algorithm for a system model whose measurement sensor performance degraded gradually due to the sensor aging (i.e., measurements are polluted with slowly-varying sensor noise covariance). That is, the slowly-varying sensor noise covariance increases linearly, exponentially, or linearly with sinusoid fluctuation. The sensor noise covariance variations are chosen to simulate certain practical engineering problems. For example, the linear, exponential increment, or linear increment with sinusoid fluctuation of sensor noise covariance represents an aging sensor with its measurement noise increase linearly, exponentially, or linearly with sinusoid fluctuation. In the simulation study, the controllability and observability of the considered system have been checked at each time step, and all simulation results are created using MATLAB R2015 with a computer equipped with an Intel Core i7 Processor and 16 GB RAM. Note that, in order to eliminate the redundancy in simulation results, only channel 1 of the multiple-input and multiple-output system presented, and channel 2 behaves similarly to channel 1.

Example: Consider the discretized version of system model matrices for the multi-input and multi-output (MIMO) linear time-varying (LTV) system is given by [Zhang, 2002]

$$\begin{cases} \mathbf{A} = \begin{bmatrix} e^{-2(t_{k+1}-t_k)} & \frac{-1}{2+\varrho}\left(e^{-(4+\varrho)t_{k+1}-4t_k} + e^{-2t_{k+1}-(2-\varrho)t_k}\right) \\ 0 & e^{-4(t_{k+1}-t_k)} \end{bmatrix}, \\ \\ \mathbf{B}_u = \begin{bmatrix} \frac{1}{2} - e^{-2(t_{k+1}-t_k)} & b_{12} \\ 0 & \frac{1}{4} - e^{-4(t_{k+1}-t_k)} \end{bmatrix}, \mathbf{C}_z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{C}_y = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}. \end{cases} \tag{3.13}$$

where $b_{12} = \frac{-1}{8+4\varrho}\left(e^{-\varrho t_{k+1}} - e^{-(4+\varrho)t_{k+1}+4t_k}\right) - \frac{1}{4-\varrho^2}\left(e^{-\varrho t_{k+1}} - e^{-2t_{k+1}-(2-\varrho)t_k}\right)$; $\mathbf{B}_\omega(k)$ is a diagonal matrix with proper dimension; $\varrho = 0.01$; t_k is the time step (discrete); and the sampling time is 0.1 second. Call back that the goal is to estimate the slowly-varying sensor noise covariance for system (3.13).

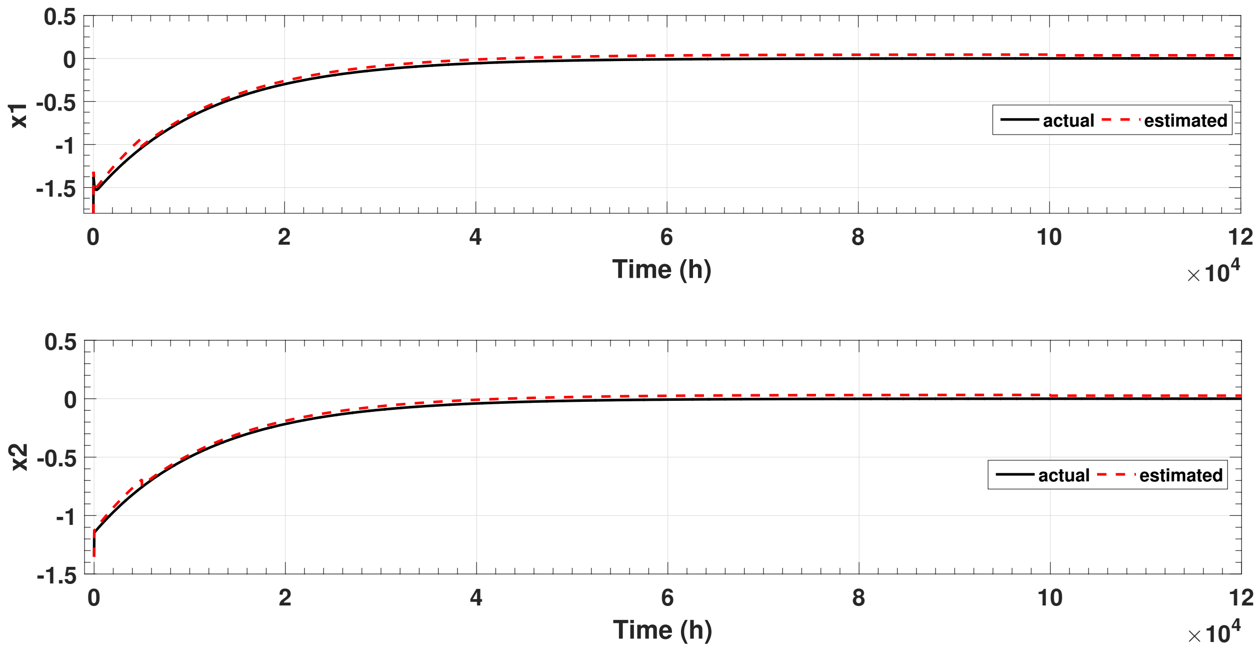


Figure 3.4: Actual and estimated states of system (3.13) with 400 samples as an estimation window length, 0.9975 forgetting factor value and linearly increased sensor noise covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is linear in state magnitude. The solid and dotted lines represent the actual and estimated state, respectively.

Fig. 3.4 shows the actual and estimated states of system (3.13) considering sensor noise covariance has linear increment with rate $9 * 10^{-4}$; estimation window size is 400 samples and forgetting factor φ is 0.9975. It is clear that the designed filter states converge to actual ones in a reasonable time. Note that the selection of estimation window size $N = 400$ will be explained later in this section.

Fig. 3.5 and Fig. 3.6 consider system (3.13) with $N = 400$ as an estimation window size and show the the actual and estimated sensor noise covariances with linear and exponential increment, respectively, for channel 1 of system (3.13).

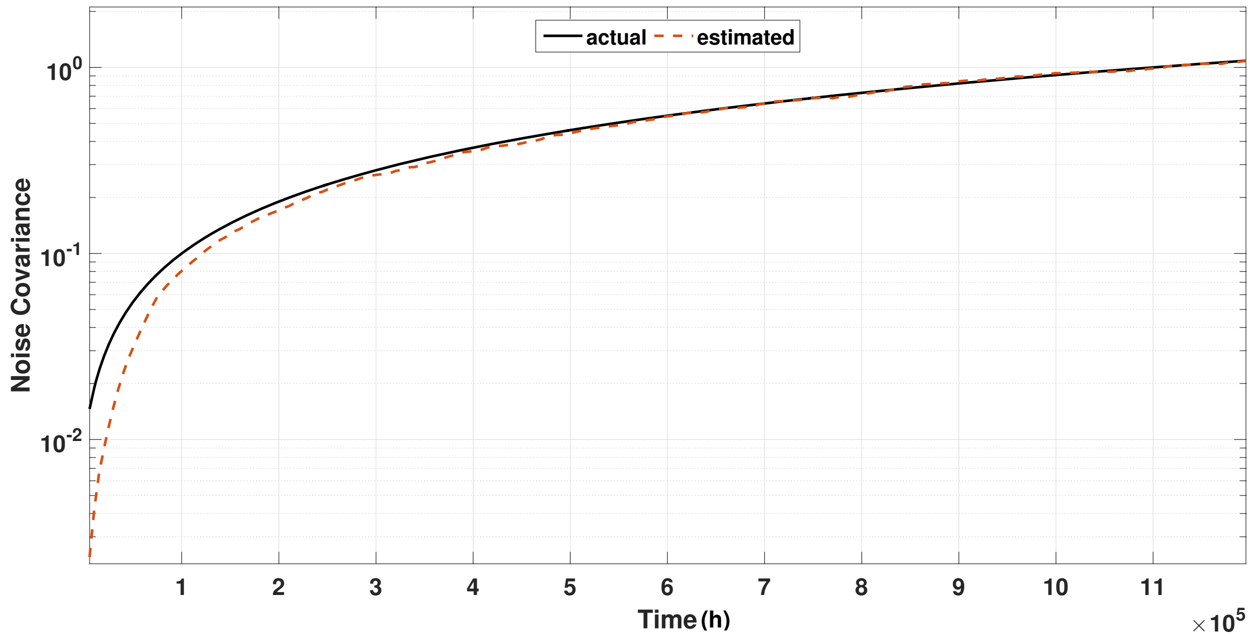


Figure 3.5: Actual and estimated sensor noise covariances for channel 1 of system (3.13) with 400 samples as an estimation window length, 0.9975 forgetting factor value and linearly increased sensor noise covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is logarithmic in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively.

Fig. 3.7 shows the actual and estimated sensor noise covariances with linearly increased covariance perturbed by a sinusoid signal for channel 1 of system (3.13). Solid and dotted-lines represent the actual and estimated sensor noise covariances, respectively. The estimated sensor noise covariances show a reasonable tracking performance for the actual ones with linear, exponential, and linear with sinusoid fluctuation cases. Indeed, for the exponential increment case, the estimated covariance matrix show good tracking performance when the slope is close to that of linear case and it deviates when the slope increases. Similarly, for the the linearly increased noise covariance with the sinusoid fluctuation, the estimated value tracks the actual one with small estimation errors. In other words, the proposed algorithm works well under a slow rate of variations for this class of noise covariances.

Fig. 3.8 and Fig. 3.9 show the simulation results of the linearly increased noise covariance with $N = 50$ and $N = 800$ samples as an estimation window lengths, respectively.

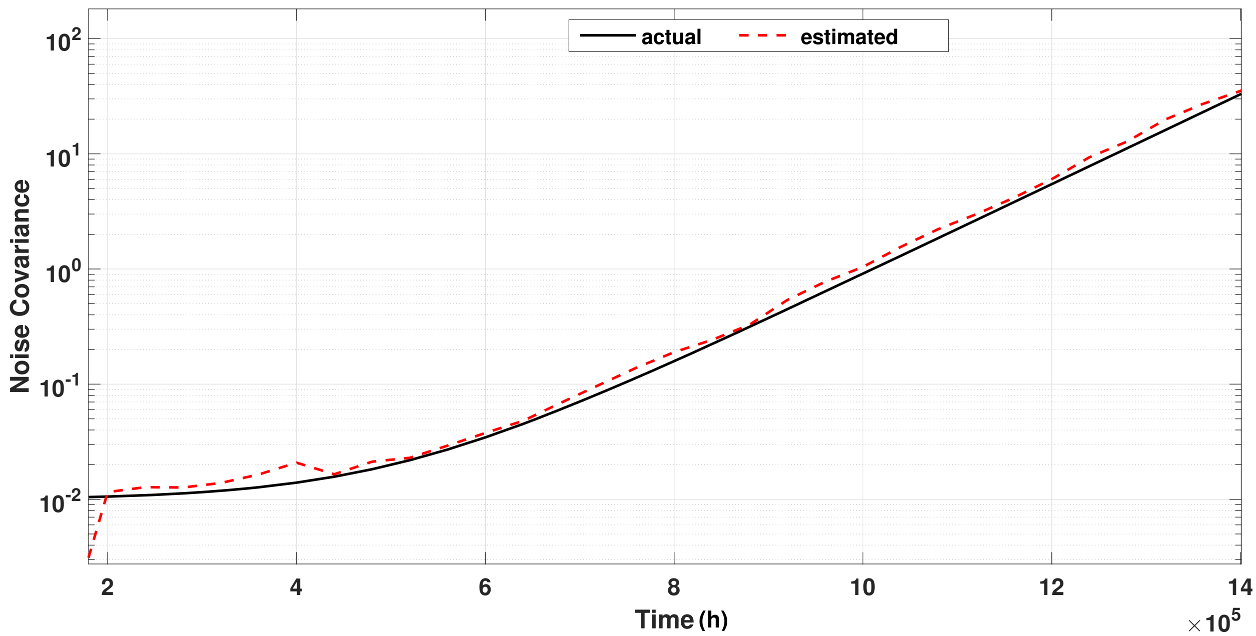


Figure 3.6: Actual and estimated sensor noise covariances for channel 1 of system (3.13) with 400 samples as an estimation window length, 0.9975 forgetting factor value and exponentially increased sensor noise covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is logarithmic in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively.

It is clear that when the estimation window length is set within certain bounds, the proposed algorithm shows a reasonable convergence. On the other hand, when the estimation window length is set outside that region, the proposed algorithm diverges (see Fig. 3.8 and Fig. 3.9). Also, the small the window length, the quick the divergence. Note that the proposed algorithm assumes that the noise covariance is fixed over the estimation window. With a short estimation window length, the estimation error at the end of estimation window will be relatively large since the estimation may not have enough time to converge, which could lead to a diverged estimation; on the other hand, with a long estimation window length, the noise covariance estimation error at the end of estimation window can be significantly reduced since there is enough time for the Kalman filter to converge, however, the assumption of constant measurement noise covariance is no long true, which could lead to large estimation error (diverged estimation). As a summary, the estimation window should be selected adequately (not too short and not too long).

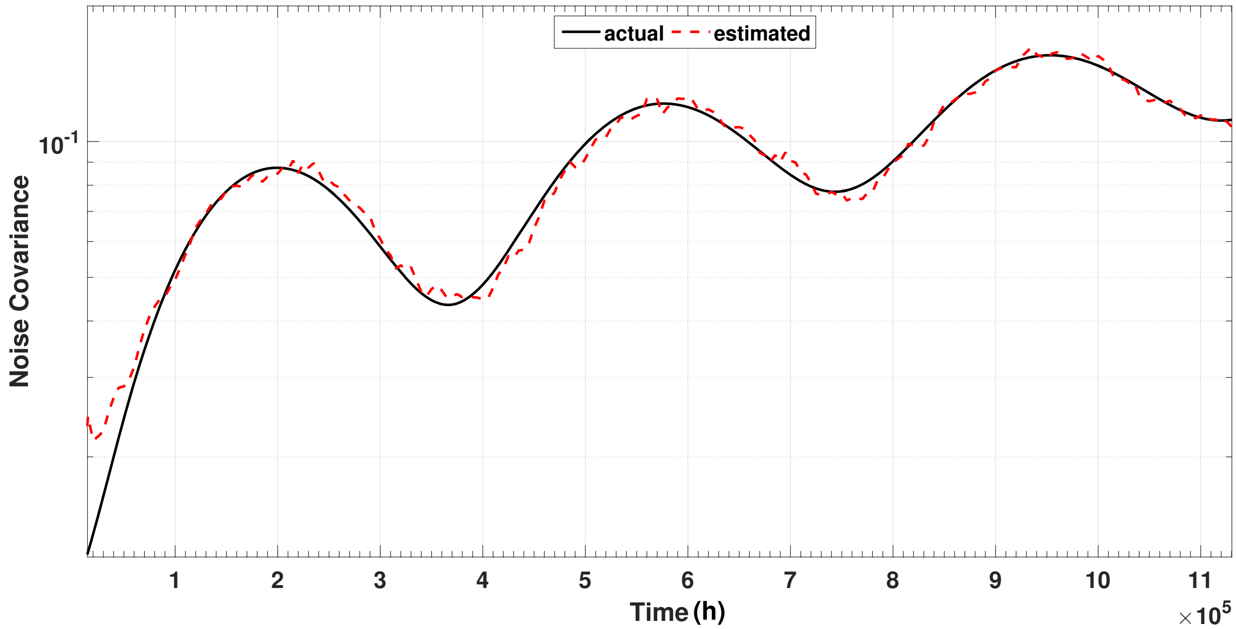


Figure 3.7: Actual and estimated sensor noise covariances for channel 1 of system (3.13) with 400 samples as an estimation window length, 0.9975 forgetting factor value and linearly increased sensor covariance perturbed by the sinusoid signal with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is logarithmic in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively.

Different estimation window lengths were tested through the simulations. Fig. 3.10 shows different estimation window lengths and corresponding noise covariance estimation errors for channel 1 of system (3.13) with a linearly increased noise covariance rate of $9 * 10^{-4}$. The case with estimation window length of 400 samples shows the smallest noise covariance estimation error, and Table 3.1 list the estimation window size (time window length) and corresponding sensor noise covariance estimation error for channel 1 of system (3.13) with the noise covariance rate of $9 * 10^{-4}$, where the noise covariance estimation error is calculated at the last time epoch of simulation run. Therefore, the estimation window length N is set to 400 and the upper and lower bounds are set to 200 and 500, respectively. It is advised to note that the upper and lower bounds of the estimation window are found under the assumption of slowly-varying sensor noise covariance.

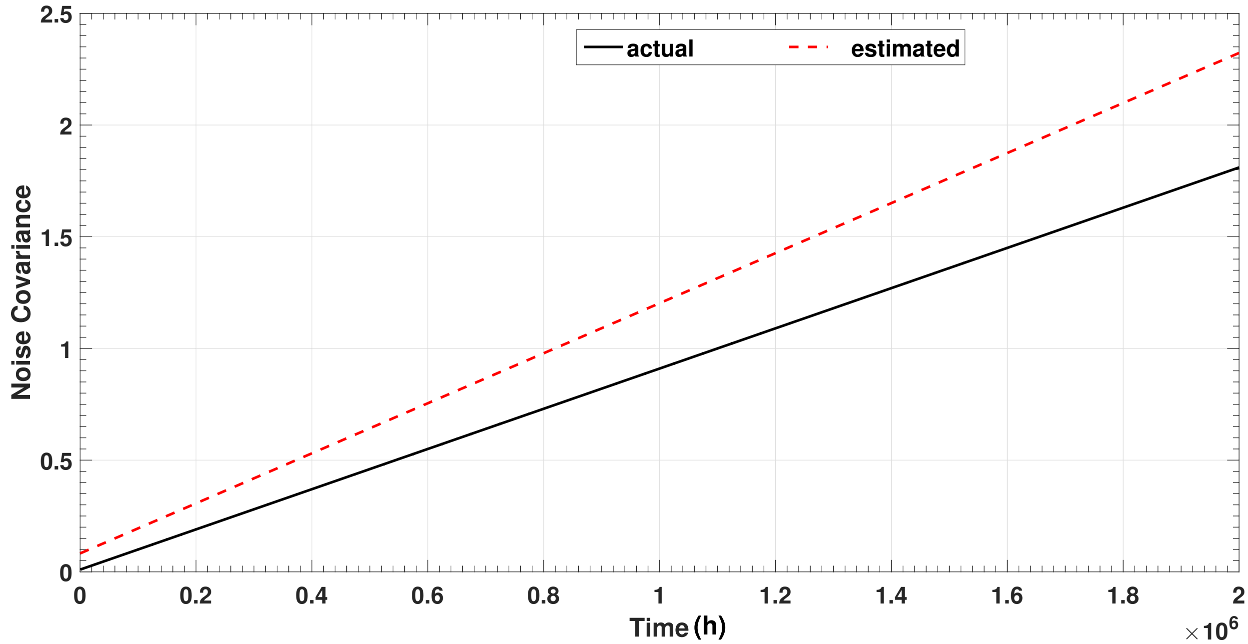


Figure 3.8: Actual and estimated sensor noise covariances for channel 1 of system (3.13) with 50 samples as an estimation window length, 0.98 forgetting factor value and linearly increased sensor noise covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is logarithmic in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively.

In contrast, when the noise covariance increase slowly and suddenly it increases very fast (changes abruptly), the proposed algorithm fails to estimate the noise covariance and the estimated noise covariance diverge. That is, the fast increment of noise covariance requires the upper bound of the estimation window size to be reduced correspondingly and that could lead to overlap with lower bounds, resulting in failed (diverged) estimation of the noise covariance for the proposed algorithm. Fig. 3.11 shows the forgetting factor value verses the noise covariance matrix estimation error for channel 1 of system (3.13) and noise covariance increases linearly with variation rate is $9 * 10^{-4}$ and estimation window size 400 samples. It is clear that forgetting value around $\varphi = 0.9975$ associated with lowest noise covariance matrix estimation error considering system (3.13) and noise covariance increases linearly with variation rate is $9 * 10^{-4}$ and estimation window size 400 samples. This matches with analytical equation, mentioned in forgetting factor value selection part last section.

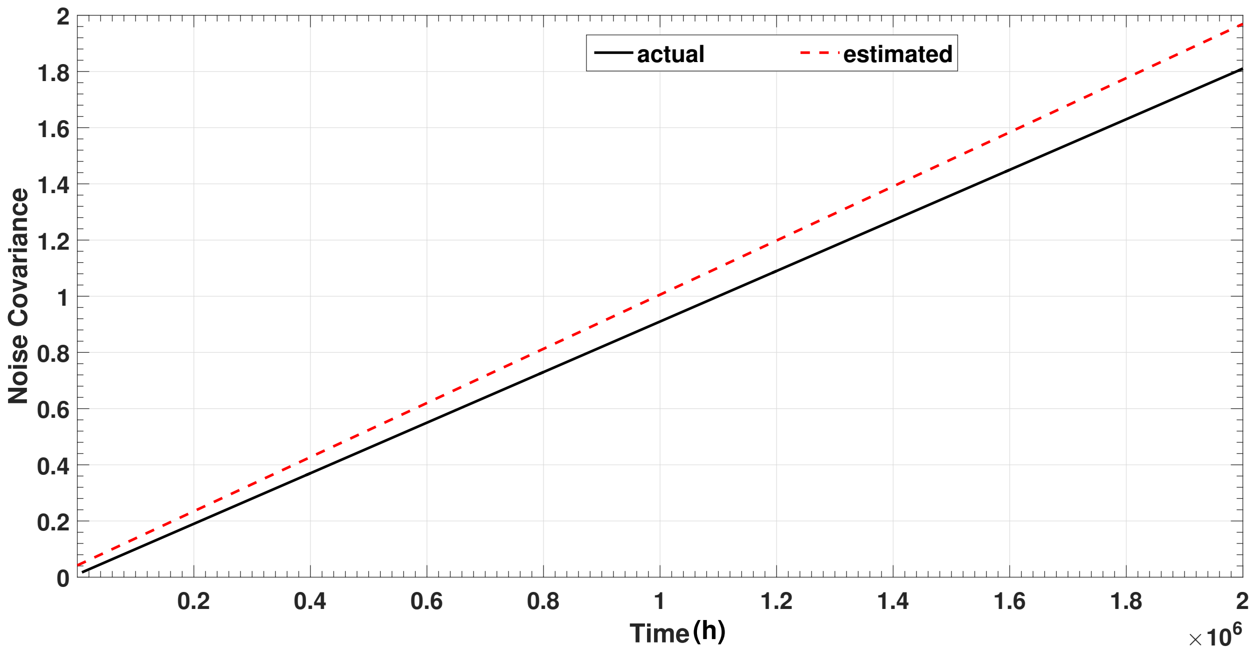


Figure 3.9: Actual and estimated sensor noise covariances for channel 1 of system (3.13) with 800 samples as an estimation window length, 0.99875 forgetting factor value and linearly increased sensor noise covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is linear in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively.

As a summary, the proposed algorithm shows its capability of detecting sensor performance degradation due to sensor aging by estimating the slowly-varying sensor noise covariance matrix under different covariance variation functions when the estimation window length is properly selected.

The proposed sensor performance of degradation detection, is compared with that presented in Refs. [Roman et al., 2003], and [Enescu et al., 2002] using Kalman filter with a recursive estimation techniques to estimate the noise covariance. Fig. 3.12 shows the actual and estimated sensor noise covariance matrix for channel 1 with 400 samples estimation window length, 0.9975 forgetting factor value and linearly increased sensor covariance with increment rate $9 * 10^{-4}$. The upper sub-figure represents the results of the proposed algorithm in this work and lower sub-figure represents the results of the algorithm used in Refs. [Roman et al., 2003], and [Enescu et al., 2002]. It is clear that the proposed algorithm in this work converges to the actual noise covariance while algorithm

Table 3.1: Estimation window size (time window length) and corresponding sensor noise covariance estimation error considering channel 1 of the system defined in (3.13) and noise covariance increases linearly with variation rate is $9 * 10^{-4}$. The noise covariance estimation error is calculated at the last time epoch of simulation run.

Estimation Window Size (N)	Noise Covariance Estimation Error (%)
50	50%
400	2%
800	17%

used in Refs. [Roman et al., 2003], and [Enescu et al., 2002] fails to converge. Moreover, the proposed filter is less sensitive to the initial value of the state and its estimation error covariance due to the iterative manner of the proposed algorithm, while algorithm in Refs. [Roman et al., 2003], and [Enescu et al., 2002] is much sensitive to initial conditions due to the recursive manner of their algorithm. That is, the proposed algorithm consider weighted innovation information at each estimation window while the other algorithm use recursive one without weighting. The computation performance of the proposed algorithm can be indicated by the time consumed by CPU for one iteration of noise covariance estimation and it is found to be 0.0087569 sec while 0.1304 sec for Refs. [Roman et al., 2003], and [Enescu et al., 2002] algorithm. The computer is equipped with an Intel i7 2.6 GHz processor with a sampling time of 0.1 sec. Again, the sample estimation window length is 400 samples, forgetting factor value is 0.9975 with linearly increased sensor covariance with increment rate $9 * 10^{-4}$. In addition, the computation performance of the proposed algorithm (i.e., the time consumed by CPU for one iteration of noise covariance estimation) is almost stationary in the proposed algorithm while it is exponentially for the one proposed by Refs. [Roman et al., 2003], and [Enescu et al., 2002] due to the recursive nature of their algorithm.

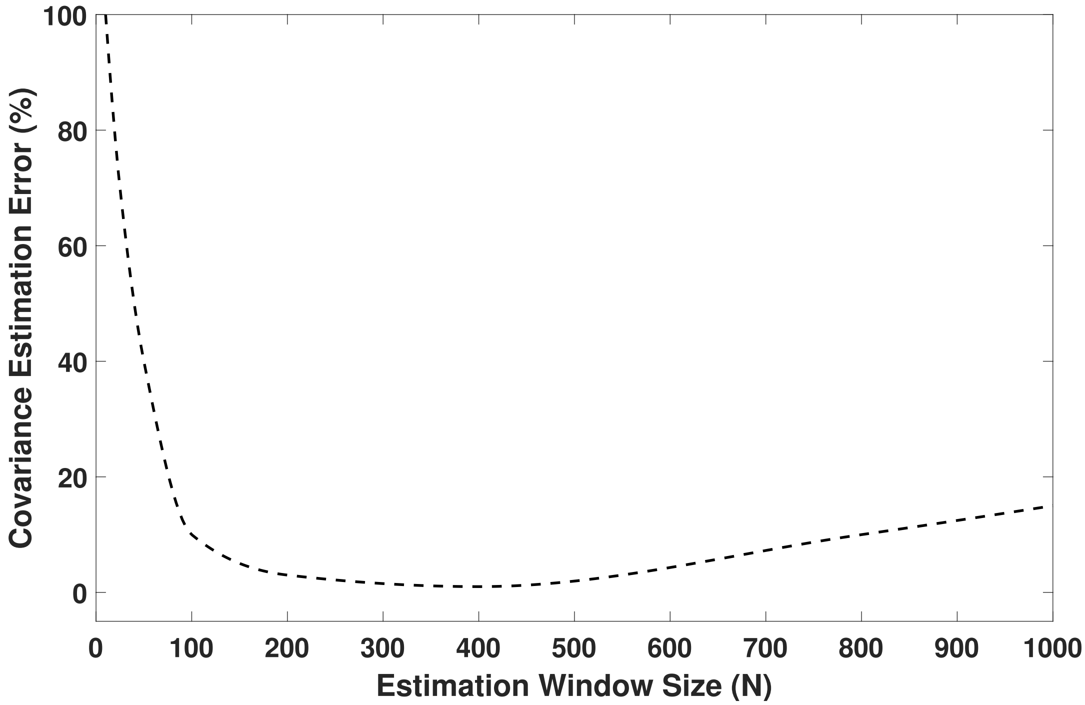


Figure 3.10: The noise covariance estimation error vs. estimation window length considering channel 1 of the system (3.13) and noise covariance increases linearly with variation rate is $9 * 10^{-4}$. The x-axis is linear in estimation window size and y-axis is linear in noise covariance estimation error.

3.5 Summary

The control systems used in industry and other fields should be safe, reliable, and stable. Degradation of sensor performance due to the sensor aging can be a crucial factor affects these system performances, reliability, and even stability. Adapting the control system to sensor performance variations helps to avoid any catastrophic consequences. This motivates monitoring sensor performance and detecting the gradual sensor performance degradation due to sensor aging. New algorithm is proposed to detect gradual sensor performance degradation due to sensor aging that is characterized by slowly-varying sensor noise covariance. In contrast to fault detection literature, the gradual sensor performance degradation is characterized by slowly-varying sensor noise covariance. In addition, most of the noise statistic identification literature consider the case with time-invariant sensor noise covariance only; while this work extends to the case with time-varying sensor noise

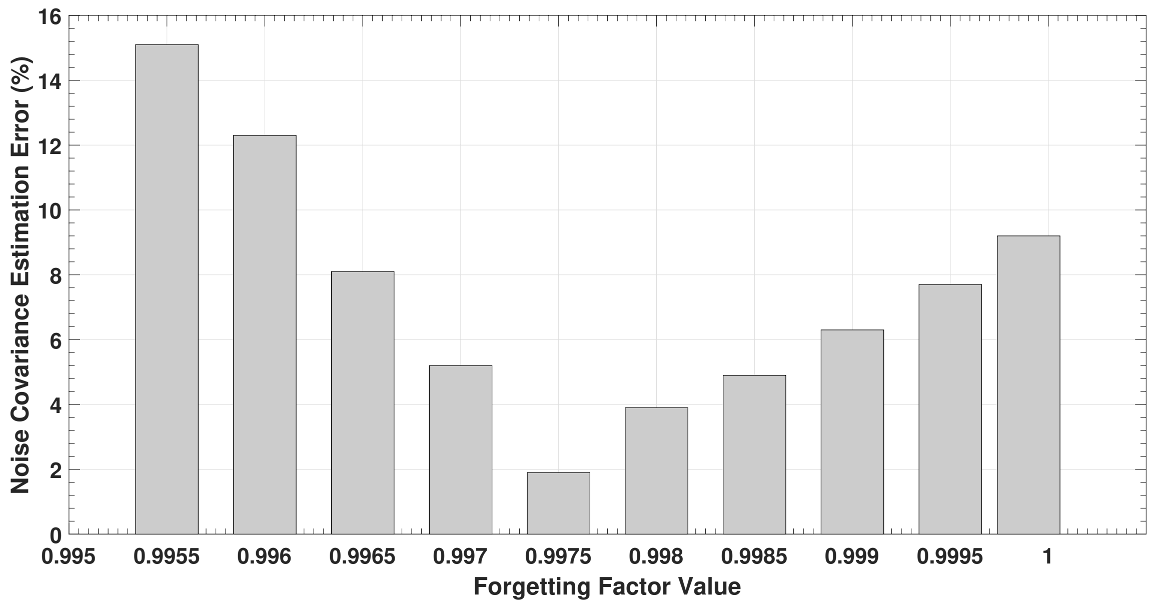


Figure 3.11: Noise covariance matrix estimation error and the forgetting factor value for channel 1 of system (3.13) and noise covariance increases linearly with variation rate is $9 * 10^{-4}$ and estimation window size 400 sample. The x-axis is linear in forgetting factor values. The y-axis is linear scale and represents the noise covariance estimation error.

covariance due to the sensor aging. That is, the gradual-degradation of sensor performance due to the sensor aging is characterized by the slowly-varying sensor measurement noise covariance. Additionally, this work incorporates the innovation vector sequence quality information with the weighted measurements, for estimating the slowly-varying sensor noise covariance. The proposed algorithm invests adaptive filtering theory with the covariance-matching technique and the information about the quality of the weighted innovation vector sequence for the estimation process. Moreover, the proposed algorithm is suitable for online applications due to the low computation load and iterative manner; and is less sensitive to initial conditions (i.e., less sensitive to initial state and its initial estimation error covariance matrix). Also, use of weighted information of the innovation vector sequence reduces computational load, and the exponentially weighted estimation window improves the estimation accuracy and makes it feasible for online applications. Consequently, the proposed algorithm could also be used as an online fault identification technique. Therefore,

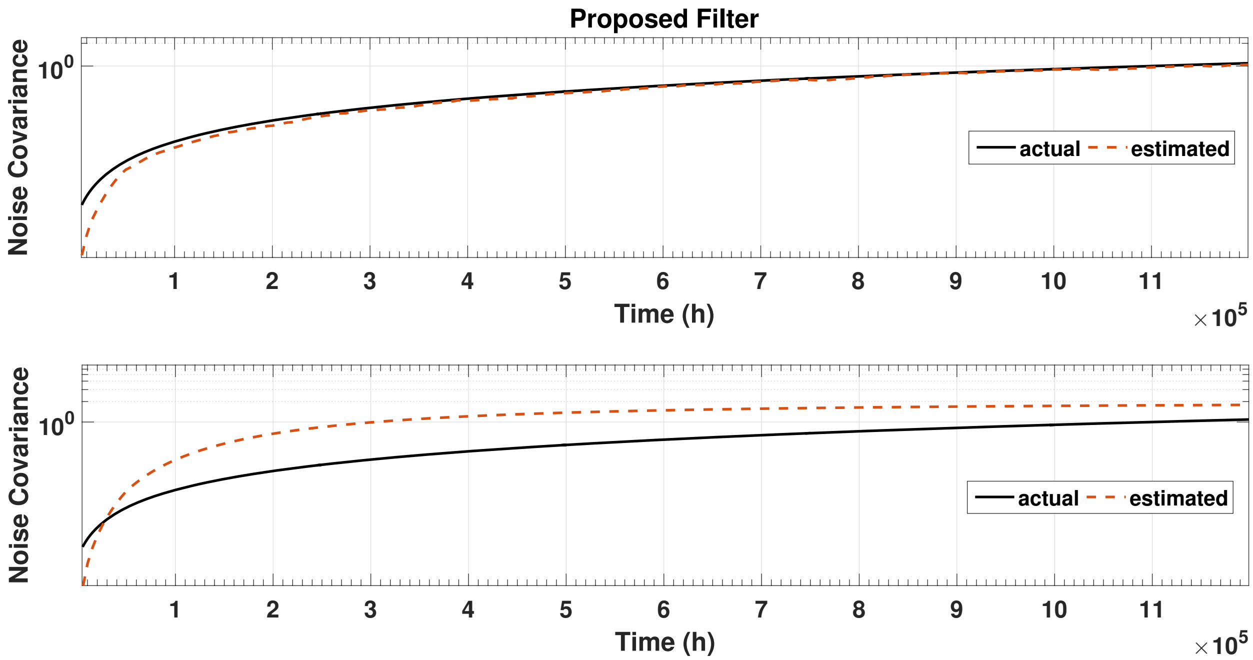


Figure 3.12: Actual and estimated sensor noise covariance matrix for channel 1 of system (3.13) with 400 sample estimation window length, 0.9975 forgetting factor value and linearly increased sensor covariance with increment rate $9 * 10^{-4}$. The x-axis is linear in time and y-axis is logarithmic in noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively. The upper sub-figure represents the result of the proposed algorithm and lower sub-figure represents the result of the algorithm used in Refs. [Roman et al., 2003, Enescu et al., 2002].

the proposed algorithm could be used as an online sensor health-monitoring and fault detection technique which is particularly important for safety-critical applications. The convergence of the proposed algorithm is demonstrated through a simulation study and shows a reasonable rate of convergence, improved estimation accuracy and reduced computation load. As a summary, the proposed algorithm can estimate the slowly-varying unknown sensor noise covariance, assuming that the estimation window is properly chosen. Next chapter demonstrate the algorithm to detect and identify the abrupt or intermittent sensor failure.

Chapter 4

Online Sensor Performance Monitoring and Fault Detection for Discrete Linear Parameter Varying Systems

4.1 Introduction

Whatever the application realm of interest may be, a sensor failure remains an unwelcome event causing an undesirable perturbation in the normal operation of a system, with multifarious adverse effects such as loss efficiency, productivity, reliability and profitability for several industries. As mentioned earlier, to minimize the possibility of unexpected failures, control systems need to enhance its ability of fault detection. Sensor fault could be caused by gradual-degradation of sensor performance (which is the focus of Chapter 3) or abrupt (intermittent) performance change (which is the focus point of this chapter), and it impacts system performance, stability, and reliability.

The recent abrupt fault detection literature focus on the effect of sensor fault to the system performance in many application fields, such as combustion engine [Isermann, 2005], nonlinear systems [Zhang et al., 2002], sewer system [Ingimundarson et al., 2009], induction machines [CusidOCusido et al., 2008], wind turbines [Wei et al., 2010], satellite network [Wu and Saif, 2007], vehicle steering system [Arogeti et al., 2012], aircraft control [Samara et al., 2008], unmanned vehicles [Abbaspour et al., 2017], building services systems [Wang and Xiao, 2004], fluid power systems [Niksefat and Sepehri, 2002], vehicle steering system [Arogeti et al., 2012], telecommunication system [Troitzsch et al., 2016], grid system [Xia et al., 2017], sensor abrupt fault using wavelet technique [Zhang and Yan, 2001], immune system [Laurentys et al., 2010], and vehicle active suspension system [Chamseddine and Noura, 2008]. Different methods were used to model, detect, and identify sensor faults. The commonly used approaches model the sensor fault as an additive or multiplicative term in the measurement equation, augment the fault terms with states, and use filter (or observer) to estimated the augmented states, a combination of system states and fault terms. Also, the sensor fault is modeled as an additive or multiplicative term and compared with the residual signal for a given threshold to detect the faulty sensor. The virtual sensor technique was used to mask the faulty sensor measurement(s) and keep the system within its nominal performance [Rotondo et al., 2014].

In contrast to others, this work assumes that the sensor performance shift can be characterized by its measurement noise covariance variation. That is, the gradual-degradation of sensor performance due to sensor aging is characterized by the slowly-varying sensor measurement noise covariance, and the sensor abrupt (or intermittent) failures are characterized by an abrupt (or intermittent) change of sensor measurement noise covariance. As a result, a sensor performance monitoring system is required to detect the slowly and/or abruptly varying sensor performance.

In Chapter 3 (see Ref. [Madhag and Zhu, b]), an algorithm was developed to detect the gradual sensor performance degradation, that is estimating the slowly-varying sensor noise covariance. An adaptive Kalman filter with the covariance-matching technique is exploited for online estimating slowly-varying sensor measurement noise covariance. In Chapter 3, it has been shown that the proposed algorithm can only handle the slowly-varying covariance case, and it cannot deal with fast-varying or sudden change in noise covariance. To match with the goal of online monitoring the sensor performance and detect the sudden change in performance a sensor abrupt (or intermittent) failure detection method is proposed in this Chapter. The proposed abrupt change detection method is based on the calculated change of estimated sensor measurement noise covariance, assuming that the failure can be characterized by the change of the estimated sensor measurement noise covariance. To be more specific, the change is quantified by the distance between the sensor measurement noise covariances over the two estimation windows before and after the change occurs. Computing the distance between two covariances can be realized by computing the distance between the sample windows [Brockwell and Davis, 2016]. Since each sample is a vector in a multi-dimensional Euclidean space, a natural distance measure between a pair of vectors is their Euclidean distance [Bernstein, 2005]. This Chapter proposes a technique for detecting abrupt (or intermittent) change based on the Euclidean distance of estimated covariances between two sliding estimation windows.

In this Chapter the primary goal is to detect and identify the abrupt (or intermittent) change of sensor measurement noise covariance using a memory-based change detection method. On the other hand, in Chapter 3 the primary gaol was to detect the gradual sensor performance degradation, that is estimating the slowly-varying sensor measurement noise covariance using a modified adaptive Kalman filter with the covariance-matching technique.

In contrast to the published literature (e.g.,[Madhag and Zhu, b]), this Chapter focus on abrupt sensor failure for discrete linear time-varying (DLTV) systems with application to discrete linear parameter-varying (LPV) systems as a special case. The proposed algorithm is suitable for online applications due to its iterative manner without making any assumption on data statistical distribution before and after the sensor performance change and with low computational load; for more details see [Madhag and Zhu, c]. Consequently, the proposed algorithm could be used as an online sensor performance monitoring scheme.

The main contributions of this Chapter is the sensor failure is characterized by an abrupt or intermittent change of sensor measurement noise covariance. The proposed algorithm is able to monitor sensor performance, detect sensor fault, and identify the sensor failure in real-time. In addition, the proposed sensor performance monitoring algorithm can be applied to discrete LPV systems as a special case of DLTV systems. Moreover, the sensor performance degradation due to sensor aging is modeled as a slow change of sensor measurement noise covariance.

This Chapter is organized as follows. Sub-section 4.2 provides an overview of the system model and related assumptions. Sub-section 4.3 describes the proposed approach for gradual-degradation of sensor performance detection and presents a memory-based technique for detecting sensor abrupt (or intermittent) failure. A discrete LPV system model is presented in sub-section 4.5 as a special case of the DLTV system. Sub-section 4.6 shows a simulation example used to demonstrate the capability of the proposed approach. Conclusions are summarized in Sub-section 4.7.

4.2 System Model

To describe the proposed algorithm clearly, the considered dynamic system model and the associated assumptions are reviewed first. The target discrete-time stochastic closed-loop system is shown in Fig. 4.1, where the controller (i.e., dotted line) is assumed to be known in this Chapter. The plant model is described in the following state-space representation

$$\begin{cases} \mathrm{x}(k+1) &= \mathbf{A}(k)\,\mathrm{x}(k) + \mathbf{B}_u(k)\,\mathrm{u}(k) + \mathbf{B}_\omega(k)\,\omega(k)\,, \\ \mathrm{z}(k) &= \mathbf{C}_z(k)\,\mathrm{x}(k) + \mathbf{D}_z(k)\,\mathrm{u}(k), \\ \mathrm{y}(k) &= \mathbf{C}_y(k)\,\mathrm{x}(k) + \nu(k)\,, \end{cases} \tag{4.1}$$

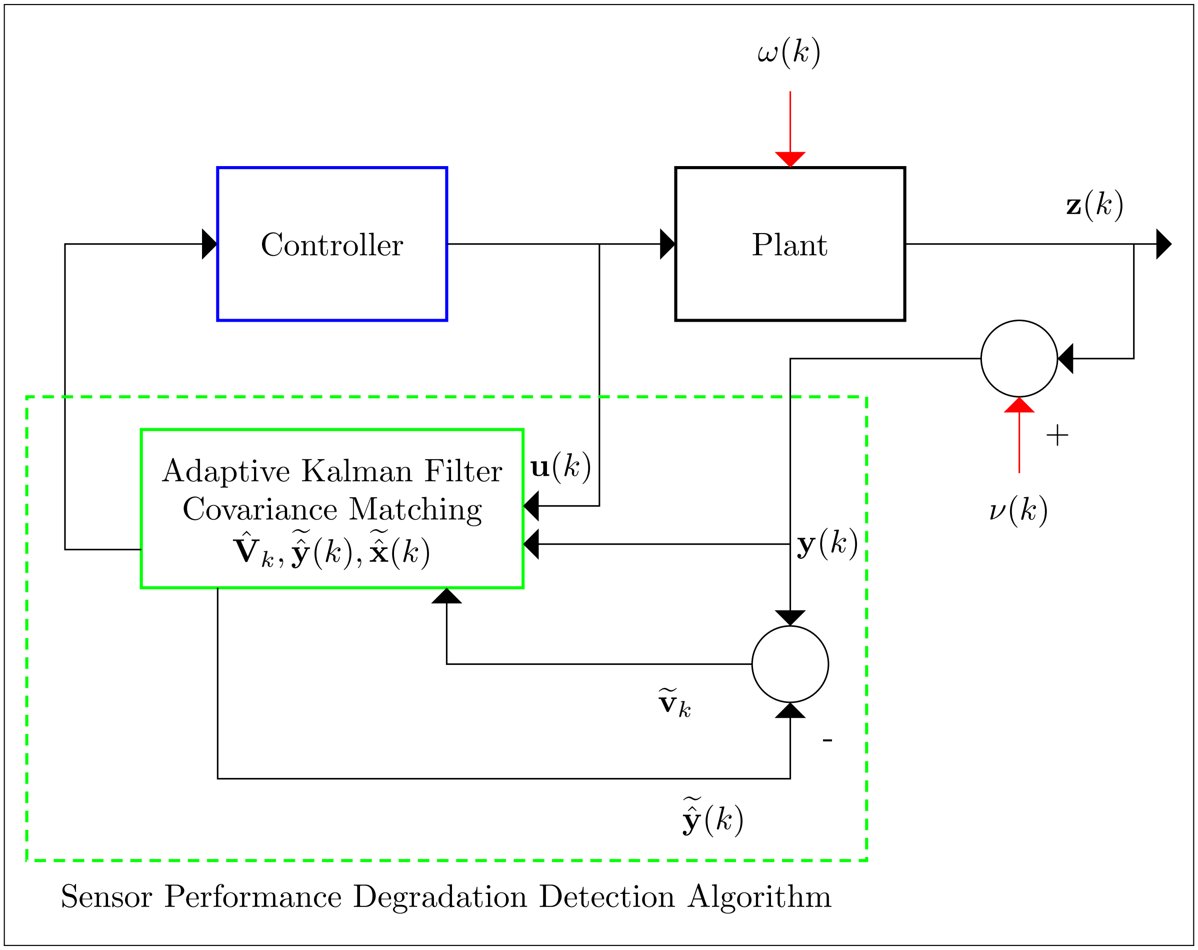


Figure 4.1: Closed-loop system. The subscript k is the sample time index; $\mathrm{x}(k)$, $\mathrm{y}(k)$ and $\mathrm{u}(k)$ are the state, measurement and input vectors, respectively; $\widetilde{\hat{\mathrm{x}}}(k)$ and $\widetilde{\hat{\mathrm{y}}}(k)$ are the estimated state and measurement vectors, respectively, via sub-optimal filter; $\widetilde{\mathbf{v}}_k$ is the innovation process ; $\omega(k)$ and $\nu(k)$ are the process and sensor measurement noises, respectively. The dotted-line represents the known controller.

where subscript k is the sample time index; $\mathrm{x}(k) \in \mathbb{R}^n$ is the state vector; $\mathrm{z}(k) \in \mathbb{R}^q$ is the controlled output vector; $\mathrm{y}(k) \in \mathbb{R}^p$ is the measured output vector; $\mathrm{u}(k) \in \mathbb{R}^m$ is the input vector; $\mathbf{A}(k) \in \mathbb{R}^{n\times n}$ is the system matrix; $\mathbf{B}_u(k) \in \mathbb{R}^{n\times m}$ is the input matrix; $\mathbf{C}_z(k) \in \mathbb{R}^{q\times n}$; $\mathbf{D}_z(k) \in \mathbb{R}^{q\times m}$ and $\mathbf{C}_y(k) \in \mathbb{R}^{p\times n}$ are the measurement matrices; $\mathbf{B}_\omega(k)$ is a diagonal matrix with proper dimension; $\omega(k)$ and $\nu(k)$ are the process and sensor measurement noises, respectively. The process and measurement noises are uncorrelated Gaussian white ones with their means and covariances defined by

$$\mathbb{E}\left[\omega(k)\right] = 0, \quad \mathbb{E}\left[\omega(k)\omega(j)^T\right] = \mathbf{W}_k\, \delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \tag{4.2}$$

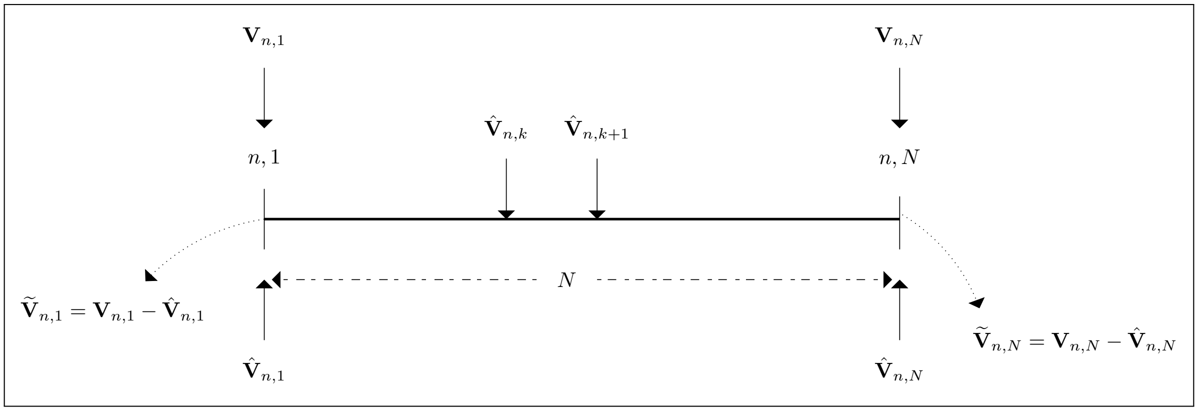


Figure 4.2: Estimation window specifications. The subscripts n and k denote the estimation window number and time within the estimation window, respectively. The length of the estimation window is denoted by N. The estimated noise covariance matrix at estimation window n and sample time k is denoted by $\hat{\mathbf{V}}_{n,k}$. The first and last time instants of the estimation window are denoted by "n, 1" and "n, N", respectively. The associated estimated noise covariances are denoted by $\hat{\mathbf{V}}_{n,1}$ and $\hat{\mathbf{V}}_{n,N}$, respectively, while any two consecutive values within the window are denoted by $\hat{\mathbf{V}}_{n,k}$ and $\hat{\mathbf{V}}_{n,k+1}$, respectively. $\mathbf{V}_{n,1}$ and $\mathbf{V}_{n,N}$ denote the actual noise covariances at first and last time instant of the estimation window.

$$\mathbb{E}\left[\nu(k)\right] = 0, \;\; \mathbb{E}\left[\nu(k)\nu(j)^T\right] = \mathbf{V}_k\, \delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \tag{4.3}$$

$$\mathbb{E}\left[\omega(k)\nu(j)^T\right] = 0, \;\; \forall\, k, j \in \mathbb{R}, \tag{4.4}$$

where $\mathbb{E}\left[\cdot\right]$ denotes the expectation operator; δ is a Kronecker Delta function (i.e., $\delta(0) = 1$ and $\delta(k) = 0; \forall\, k \neq 0$); and $\mathbf{W}_k$ and $\mathbf{V}_k$ are time-varying process and measurement noise covariance matrices, respectively. The initial state vector x(0) is assumed to have a normal distribution with zero mean ($\mathbb{E}\left[\mathrm{x}(0)\right] = 0$) and covariance ($\mathbf{P}_0 = \mathbb{E}(\mathrm{x}(0)\mathrm{x}^T(0)) > 0$). The system is assumed to be uniformity observable ans controllable. The process noise covariance matrix, $\mathbf{W}_k$, is assumed to be positive semi-definite and known at each time instant. The measurement noise covariance, $\mathbf{V}_k$, is assumed to be positive definite. In the numerical calculations, the negative eigenvalues of matrix $\mathbf{V}_k$ are replaced by small positive values. Moreover, all sensors measurement noises of all channels are assumed to be independent (i.e., measurement noise for each channel is independent on these for the rest of measurements), that is, $\mathbf{V}_k$ is a diagonal matrix.

4.3 Detecting Gradual Sensor Performance Degradation

The primary goal of this Chapter detecting sudden (fast) change of the sensor performance, which is characterized by the abrupt (or intermittent) change of sensor measurement noise covariance. Call back that the goal in Chapter 3 is to online monitor sensor performance degradation, and detect sensor failure, that is, online estimating the slowly-varying sensor measurement noise covariance due to the sensor aging. To achieve such goal, we invest the modified adaptive Kalman filter with covariance-matching technique and a memory-based change detection method for a system with measurement noise covariance varying slowly and/or abruptly.

The gradual-degradation of sensor performance due to sensor aging is characterized by its slowly-varying measurement noise covariance. An algorithm was developed by Chapter 3; also see Ref. [Madhag and Zhu, b], to online estimate the unknown slowly-varying sensor measurement noise covariance from the set of measurements based on the quality of the weighted innovation vector sequence, where an adaptive Kalman filter [Simon, 2006] with the covariance-matching technique is adopted. Note that the innovation vector is defined as the difference between the actual measurement and its predicted value [Madhag and Zhu, b], [Simon, 2006] and given by

$$\tilde{\mathbf{v}}_k = \mathrm{y}(k) - \mathbf{C}_y(k)\,\widetilde{\hat{\mathrm{x}}}(k|k-1), \tag{4.5}$$

where $\widetilde{\hat{\mathrm{x}}}(k|k-1)$ is the estimated state vector at time k based on measurement at time $k-1$ using adaptive Kalman filter [Madhag and Zhu, b]; $\mathrm{y}(k)$ is the measured vector at time k; $\mathbf{C}_y(k)$ is the measurement matrix at time k; $\tilde{\mathbf{v}}_k$ is the innovation vector; and the superscripts tilde and head denote the sub-optimal and estimated values, respectively. Indeed, to obtain a feasible estimation based on the available measurement (minimal available information) and to reduce estimation error and computational load, the last estimation is then combined with a weighted innovation vector.

The weighting coefficient φ is used as a "forgetting factor", and it is adaptively adjusted to improve the estimation accuracy by put more weight for the recent innovation data and less weight for the old one, which improve the trade-off between estimation bias and variance; see Chapter 3 for details. The weighted innovation vector estimate can be rewritten as follows

$$\bar{\mathbf{v}}_N = \frac{1}{N}\left[\sum_{k=1}^{N} \varphi^{N-k}\, \tilde{\mathbf{v}}_k\right], \tag{4.6}$$

where N is the number of time samples; $\varphi \in (\epsilon, 1]$ $(0 < \epsilon < 1)$ is the forgetting factor for the innovation data vector in the past; $\bar{\mathbf{v}}_N$ is the innovation vector estimate with the forgetting factor. Then, the estimate of the sensor measurement noise covariance at estimation window n with sample time $k = N$ (i.e., at the end of current estimation window) can be written as

$$\hat{\mathbf{V}}_{n,N} = \frac{1}{N-1}\sum_{k=1}^{N}\left[(\varphi^{N-k}\tilde{\mathbf{v}}_k - \bar{\mathbf{v}}_N)(\varphi^{N-k}\tilde{\mathbf{v}}_k - \bar{\mathbf{v}}_N)^T - \left(\frac{N-1}{N}\right)\mathbf{C}_y(k)\widetilde{\hat{\mathbf{P}}}_{k|k-1}\mathbf{C}_y^T(k)\right], \tag{4.7}$$

where $\tilde{\mathbf{v}}_k$ is the innovation vector; $\bar{\mathbf{v}}_N$ is innovation vector estimate; $\widetilde{\hat{\mathbf{P}}}_{k|k-1}$ is state estimation error covariance; $\hat{\mathbf{V}}_{n,N}$ is the estimated noise covariance matrix at the end of estimation window n with forgetting factor φ. The noise covariance estimation window specifications are shown in Fig 4.2. Note that, by considering the forgetting factor for the innovation data vector, the estimation window is exponentially weighted providing improved estimation accuracy; see Chapter 3 for more details. In addition, it is advised to see Chapter 3 or [Madhag and Zhu, b] for details of obtaining equation (4.7). It has been shown in Chapter 3 that the proposed algorithm is able to handle the slowly-varying covariance only and is not suitable for the case with fast-varying or sudden change of noise covariance. The next sub-section addresses the detection technique of sensor measurement noise covariance with abrupt or intermittent change.

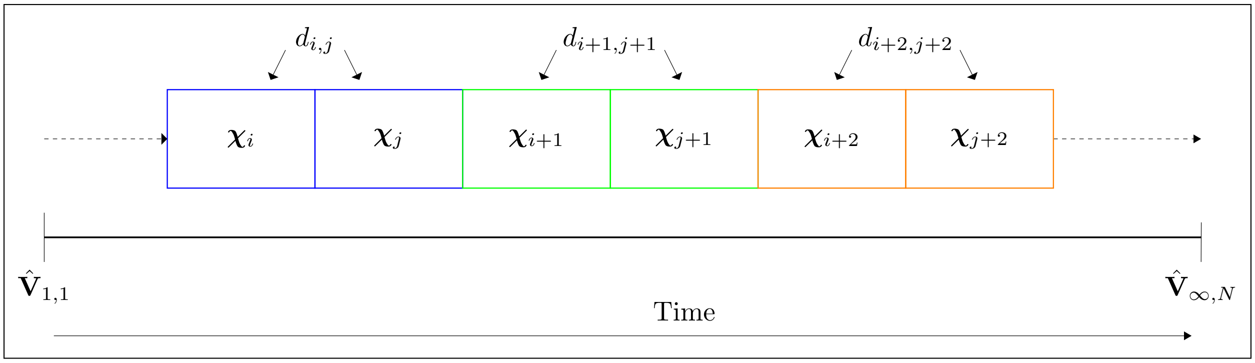


Figure 4.3: A snapshot of two sliding estimation windows. The symbol $\hat{\mathbf{V}}_{1,k}$ represents the sensor estimated noise covariance at estimation window number 1 and time instant k with $1 \leq k \leq N$; the symbol $\hat{\mathbf{V}}_{\infty,k}$ represents the sensor estimated noise covariance at estimation window number ∞ and time instant k, where in simulation study the number of noise covariance estimation windows is set to very large number; the length of the estimation window is denoted by N; and χ_i and χ_j are two consecutive sliding estimation windows of size m.

4.4 Abrupt or Intermittent Sensor Fault Detection

Our focus at this sub-section is to online detect the sudden change in sensor measurement noise covariance from a set of measurements. Hence, a proposed memory-based technique for detecting the sudden change of sensor measurement noise covariance is proposed. The proposed change detection technique calculates the change of the estimated sensor measurement noise covariance, assuming that the sensor failure is characterized by abruptly (or intermittently) changing of the estimated sensor measurement noise covariance. To be more specific, the change is quantified by the distance between the sensor measurement noise covariance matrices over the two estimation windows before and after the change. Computing the distance between two covariance matrices can be realized by computing the distance between the samples themselves [Brockwell and Davis, 2016]. Since each sample is a matrix in a multi-dimensional Euclidean space, a natural distance between the pair of matrices is their Euclidean distance [Brockwell and Davis, 2016]. The proposed change detection technique of sensor measurement noise covariance calculates the distance between two sliding windows of the estimated sensor measurement noise covariance matrices over two sequential estimation windows. That is, the two sliding estimation windows with index i and size m of the estimated noise covariances are given by

$$\begin{cases} \mathcal{X}_i &= \left[\hat{\mathbf{V}}_{n,k}, \cdots, \hat{\mathbf{V}}_{n,k+m}\right], \quad \mathcal{X}_j = \left[\hat{\mathbf{V}}_{n,k+m+1}, \cdots, \hat{\mathbf{V}}_{n,k+2m}\right], \\ \mathcal{X}_{i+1} &= \left[\hat{\mathbf{V}}_{n,k+2m+1}, \cdots, \hat{\mathbf{V}}_{n,k+3m}\right], \quad \mathcal{X}_{j+1} = \left[\hat{\mathbf{V}}_{n,k+3m+1}, \cdots, \hat{\mathbf{V}}_{n,k+4m}\right], \end{cases} \tag{4.8}$$

and Euclidean distance between the two sliding estimation windows is given by

$$d_{i,j} = \|\mathcal{X}_i - \mathcal{X}_j\|, \tag{4.9}$$

where $\|\cdot\|$ denotes the Euclidean norm; $\hat{\mathbf{V}}_{n,a}$ is the estimated noise covariance at estimation window n and time instant $a \in [i, j]$; and subscript m is the sliding estimation window size; see Figs. 4.3 and 4.4 for details. Note that, Fig. 4.3 shows the two sliding estimation windows, that used by the proposed memory-based technique for sensor failure detection, over the time horizon where the noise covariance estimation windows consecutively processed while Fig. 4.2 shows one windows specifications used to estimate the noise covariance. The proposed memory-based technique for sensor failure detection calculates $d_{i,j}$ for the pair of sliding estimation windows over the two sequential estimation windows, compares it with the one obtained in the previous step, and decides if the sensor fault occurred using an indicator function (signal) as follows

$$\mathcal{I}_{i,j} = \begin{cases} 0 & \text{if } \; d_{i,j} \leq d_{i-m,j-m} + \Upsilon_{i-m,j-m} \\ 1 & \text{otherwise} \end{cases} \tag{4.10}$$

where $\mathcal{I}_{i,j}$ is the indicator signal; $d_{i,j}$ is the Euclidean covariance distance between two sliding estimation windows; $d_{i-m,j-m}$ is the Euclidean covariance distance between two sliding estimation windows at the previous step; and $\Upsilon_{i-m,j-m}$ is a threshold specified by the confidence interval distance between the current and previous sliding estimation windows; see Ref. [Brockwell and Davis, 2016] for more details. Once the indicator signal $\mathcal{I}_{i,j}$ is set to 1, the stored previous and current ones are compared to identify the fault based on whether the change of difference is intermittent, saturated or abrupt with reduced computational cost. Therefore, the proposed change detection technique can be used to online identify the sensor failure type. The proposed algorithm for abrupt or intermittent sensor fault detection is shown in Fig.4.4.

To this point, the proposed technique combining the online estimating technique of the slowly-varying [Madhag and Zhu, b] sensor measurement noise covariance due to the sensor aging and detecting method of the abrupt change of sensor measurement noise covariance is introduced.

Lemma 15 *Considering system* (4.1) *and adaptive Kalman filter in Chapter 3 with aforementioned assumptions, the estimation of the slowly-varying sensor measurement noise covariance is given by equation* (4.7) *and an indicator of abrupt noise covariance change is given by* (4.10).

The above Lemma states the proposed algorithm for estimating the slow degradation of sensor performance and detecting abrupt sensor failure. The next step is to apply the proposed algorithm, developed for LTV systems, to the LPV systems, a special case of LTV systems.

4.5 Modeling LPV System as LTV or LTI System

The algorithm proposed for sensor performance monitoring and failure detection is developed for LTV systems. This sub-section presents the procedure to model an LPV system in the environment of an LTV system so that the proposed algorithm can be applied. Recently, LPV systems have received considerable attention in control community. Such models have been used effectively in aircraft, robotics, and process control problems [Mohammadpour and Scherer, 2012]. An LPV

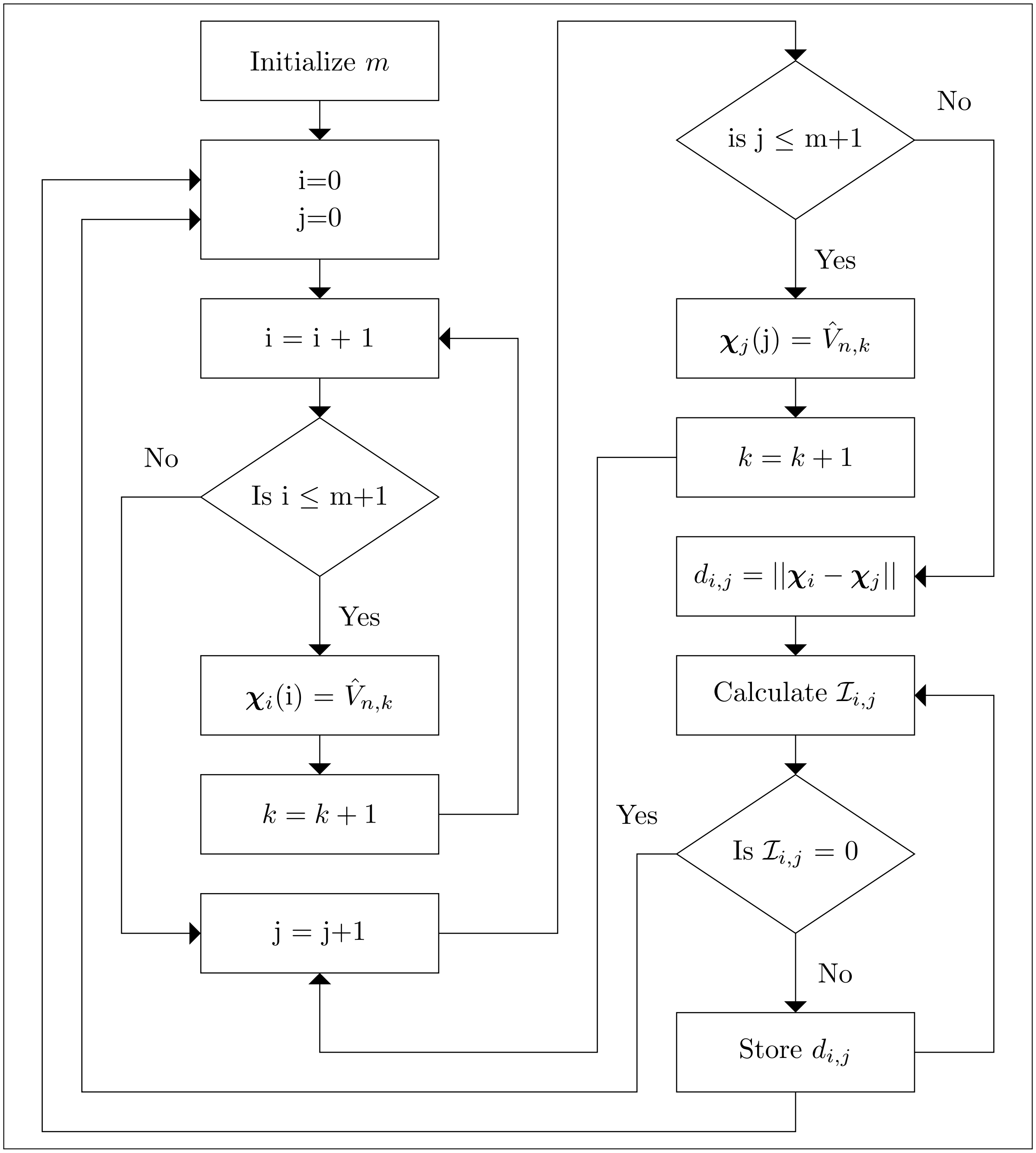


Figure 4.4: Flow chart for abrupt sensor fault detection algorithm. $\hat{\mathbf{V}}_{n,k}$ represents the sensor estimated noise covariance at estimation window number n and time instant k with $1 \leq k \leq N$; and the length of the estimation window is denoted by N; $\chi_i(i)$ and $\chi_j(j)$ are the $i-th$ and $j-th$ element of the two consecutive sliding estimation windows (i.e., χ_i and χ_j) of size m; $d_{i,j}$ denotes the Euclidean distance; and $\mathcal{I}_{i,j}$ is the indicator function (signal).

system is a linear system with parameter-dependent system matrices as functions of a measurable time-varying parameter vector belonging to a compact set. Hence, the LPV system is a linear system dependent on one or more time-varying and measurable parameters in real-time and unknown during control design. The LPV plant model could be described using the following state-space representation

$$\begin{cases} x(k+1) & = \mathbf{A}(\theta(k))\, x(k) + \mathbf{B}_u(\theta(k))\, u(k) + \mathbf{B}_\omega(k)\, \omega(k)\,, \\ z(k) & = \mathbf{C}_z(\theta(k))\, x(k) + \mathbf{D}_z(\theta(k))\, u(k), \\ y(k) & = \mathbf{C}_y(\theta(k))\, x(k) + \nu(k)\,, \end{cases} \tag{4.11}$$

where $\mathbf{A}(\theta(k))$; $\mathbf{B}_u(\theta(k))$; $\mathbf{C}_z(\theta(k))$; and $\mathbf{C}_y(\theta(k))$ are appropriately dimensioned parameter-dependent matrices; $\mathbf{B}_\omega(k)$ is diagonal matrix with proper dimension; and all assumptions of the process and measurements noises listed in Chapter 3 remains unchanged. Note that matrix $\mathbf{B}_\omega(k)$ can be parameter-dependent if the noise statistics are fully known. The time-varying scheduling parameters in equation (4.11), $\theta(k)$ are assumed to lie in a known hyper-rectangle Ω_{n_θ}, the scheduling parameters vector is given by

$$\theta(k) = \left[\theta_1(k), \theta_2(k), \cdots, \theta_{n_\theta}(k)\right]^T, \tag{4.12}$$

where n_θ is the number of scheduling parameters. The state-feedback control law is given by

$$u(k) = \mathbf{K}(\widetilde{\theta}(k))\, x(k) \tag{4.13}$$

where $\mathbf{K}(\widetilde{\theta}(k))$ is the parameter-dependent state-feedback control law gain and the scheduling parameters available for the controller (4.13) are either be inexactly measured or estimated. The measured (or estimated) time-varying scheduling parameters vector, $\widetilde{\theta}(k)$, is given by

$$\widetilde{\theta}(k) = \left[\widetilde{\theta}_1(k), \widetilde{\theta}_2(k), \cdots, \widetilde{\theta}_{n_\theta}(k)\right]^T. \tag{4.14}$$

with

$$\widetilde{\theta}_i(k) = \theta_i(k) + \vartheta_i(k), \quad \forall\, i = 1, \cdots, n_\theta,\ \forall\, k \in \mathbb{Z}_+, \tag{4.15}$$

where $\vartheta_i(k)$ represents the measurement (or estimation in case of virtual sensor) error in the i^{th} scheduling variable and $\theta_i(k)$ is its true value. The measured scheduling parameters and associated uncertainties are assumed to be independent and time-varying parameters with the following known bounds,

$$\underline{\widetilde{\theta}} \leq \widetilde{\theta}_i(k) \leq \overline{\widetilde{\theta}}, \qquad \underline{\vartheta} \leq \vartheta_i(k) \leq \overline{\vartheta}, \quad \forall\, i = 1, \cdots, n_\theta,\ \forall\, k \in \mathbb{Z}_+, \tag{4.16}$$

where $\underline{\widetilde{\theta}}$ and $\overline{\widetilde{\theta}}$; $\underline{\vartheta}$ and $\overline{\vartheta}$ are the known lower and upper bounds of the measured scheduling parameters and their measurement (or estimation) error, respectively. Note that, the values of the measured scheduling parameters are known, while the actual scheduling parameters and associated uncertainties are unknown for covariance estimation and change detection algorithm. Moreover, the measured scheduling parameters are assumed to have their variation rates in the same order of the system dynamics. Indeed, the parameter-dependent matrices for system (4.11) are depending on actual scheduling parameter vector $\theta(k)$ since it represents the physical system parameters. On the other hand, those physical parameters can only be measured with noise or estimated to be used in the controller (i.e., unknown during control design), and as a result, controller depends on measured or estimated scheduling parameter vector $\widetilde{\theta}(k)$. To elevate the calculation for modeling LPV systems in the environment of LTV systems so that the proposed algorithm can be applied for monitoring the sensor performance degradation and failure, the parameter-dependent matrices for system (4.11) are assumed to be dependent on the measured scheduling parameter vector $\widetilde{\theta}(k)$. Note that this assumption may introduce additional conservativeness to the control design but it makes the calculation simple. System (4.11) with dependence on measured scheduling parameters vector is given by

$$\begin{cases} \mathrm{x}(k+1) & = \mathbf{A}(\widetilde{\theta}(k))\ \mathrm{x}(k) + \mathbf{B}_u(\widetilde{\theta}(k))\ \mathrm{u}(k) + \mathbf{B}_\omega(k)\ \omega(k)\,, \\ \mathrm{z}(k) & = \mathbf{C}_z(\widetilde{\theta}(k))\ \mathrm{x}(k) + \mathbf{D}_z(\widetilde{\theta}(k))\ \mathrm{u}(k), \\ \mathrm{y}(k) & = \mathbf{C}_y(\widetilde{\theta}(k))\ \mathrm{x}(k) + \ \nu(k)\,, \end{cases} \tag{4.17}$$

where the matrices and parameters have the same definition as these in equation (4.11). The LPV system (4.17) is assumed to have an affine parameter dependence, for instance, the system parameter-dependent matrix can be expressed below

$$\mathbf{A}(\widetilde{\theta}_k) = \mathbf{A}_0 + \sum_{i=1}^{n_\theta} \widetilde{\theta}_i(k)\mathbf{A}_i, \tag{4.18}$$

where $\mathbf{A}_0, \mathbf{A}_i \in \mathbb{R}^{n\times n}$ are constant matrices; and $\widetilde{\theta}_i(k)$ is the i^{th} element of the scheduling parameter vector in equation (5.3) for $i = 1, \cdots, n_\theta$. The other parameter-dependent system matrices in equation (4.17) have the same structure as equation (4.18). Since LPV systems belong to a subset of linear time-varying (LTV) systems, it is reasonable to consider LPV systems as a special case of the LTV systems. Follow the technique used in [Lopes dos Santos et al., 2011], the LPV system representation is modeled as an LTV system, for instance, the parameter-dependent system matrix in equation (4.17) can also be written in the form

$$
\begin{aligned}
\mathbf{A}(\widetilde{\theta}_k)\ \mathrm{x}(k) &= \left(\mathbf{A}_0 + \sum_{i=1}^{n_\theta} \widetilde{\theta}_i(k)\mathbf{A}_i\right)\ \mathrm{x}(k) \\
&= \mathbf{A}_0\ \mathrm{x}(k) + \mathbf{A}_p\left[\widetilde{\theta}_k \otimes \mathrm{x}(k)\right], \\
&= \widehat{\mathbf{A}}_p\ \ \mathrm{x}(k)^p
\end{aligned}
\tag{4.19}
$$

where "⊗" is the Kronecker product operator; $\widehat{\mathbf{A}}_p = \left[\mathbf{A}_0 \quad \mathbf{A}_p\right]$; $\mathbf{A}_p = \left[\mathbf{A}_1\ \ \mathbf{A}_2 \cdots \mathbf{A}_{n_\theta}\right]$; $\mathrm{x}(k)^p = \left[\mathrm{x}(k)\ \ \widetilde{\theta}_k \otimes \mathrm{x}(k)\right]^T$; $\mathrm{u}(k)^p = \left[\mathrm{u}(k)\ \ \widetilde{\theta}_k \otimes \mathrm{u}(k)\right]^T$; and $\widetilde{\theta}_k$ is the scheduling parameter vector defined in equation (5.3). In the same manner, applying the procedure used in equation (4.19) to other parameter-dependent matrices in equation (4.17) $\left(i.e., \widehat{\mathbf{B}}_u^p, \widehat{\mathbf{D}}_z^p, \widehat{\mathbf{C}}_z^p, \widehat{\mathbf{C}}_y^p\right)$; with $\widehat{\mathbf{B}}_\omega = \left[\mathbf{B}_\omega(k)\ \ \mathbf{0}\right]$, and $\omega^p(k) = \left[\omega(k)\ \ \mathbf{0}\right]^T$, and $\nu^p(k) = \left[\nu(k)\ \ \mathbf{0}\right]^T$, then, the LPV system (4.17) becomes

$$
\begin{cases}
\mathrm{x}(k+1) & = \widehat{\mathbf{A}}_p\ \ \mathrm{x}(k)^p + \widehat{\mathbf{B}}_u^p\ \ \mathrm{u}(k)^p + \widehat{\mathbf{B}}_\omega\ \ \omega^p(k), \\
\mathrm{z}(k) & = \widehat{\mathbf{C}}_z^p\ \ \mathrm{x}(k)^p + + \widehat{\mathbf{D}}_z^p\ \ \mathrm{u}(k)^p, \\
\mathrm{y}(k) & = \widehat{\mathbf{C}}_y^p\ \ \mathrm{x}(k)^p + \ \nu^p(k)\,,
\end{cases}
\tag{4.20}
$$

where all parameters are defined earlier. As a result, the LPV system (4.17) is modeled as an LTV system (4.20). Consequently, the proposed algorithm for sensor performance monitoring and failure detection can be applied. Next, a simulation investigation example is used to validate the proposed approach for estimating the slowly-varying sensor measurement noise covariance and detecting the abrupt change of sensor measurement noise covariance.

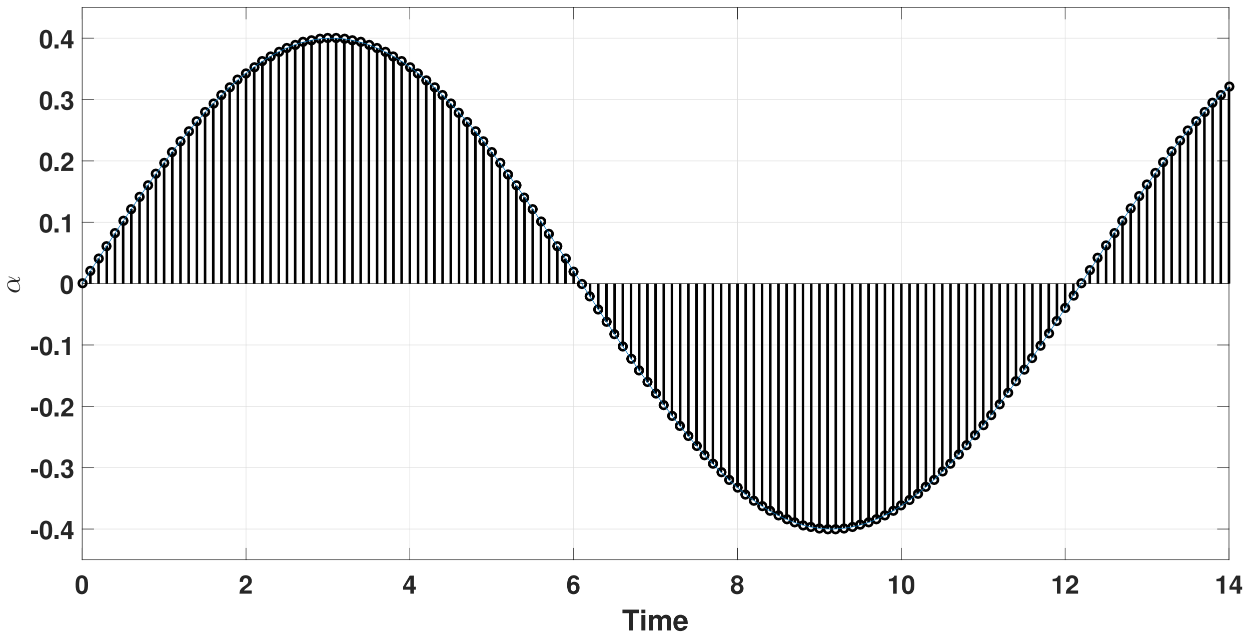
4.6 Simulation Results and Discussion

The objective of conducting simulation study using a numerical example in this sub-section is to demonstrate the ability of the proposed algorithm for online monitoring the sensor performance deterioration and detecting abrupt sensor failure for LPV systems with system parameters measurements polluted by measurement noise. The variation of sensor measurement noise covariance is chosen to simulate practical engineering problems. For example, the slowly-varying sensor noise covariance could represent an aging sensor with its measurement noise (i.e., measurement error) increases slowly. The abrupt change of sensor measurement noise covariance could represent a faulty sensor with its measurement noise change abruptly or intermittently. In the simulation study, the observability and controllability of the considered system have been checked at each time step, and all simulation results are created using MATLAB R2015 with a computer equipped with an Intel Core i7 Processor and 16 GB RAM.

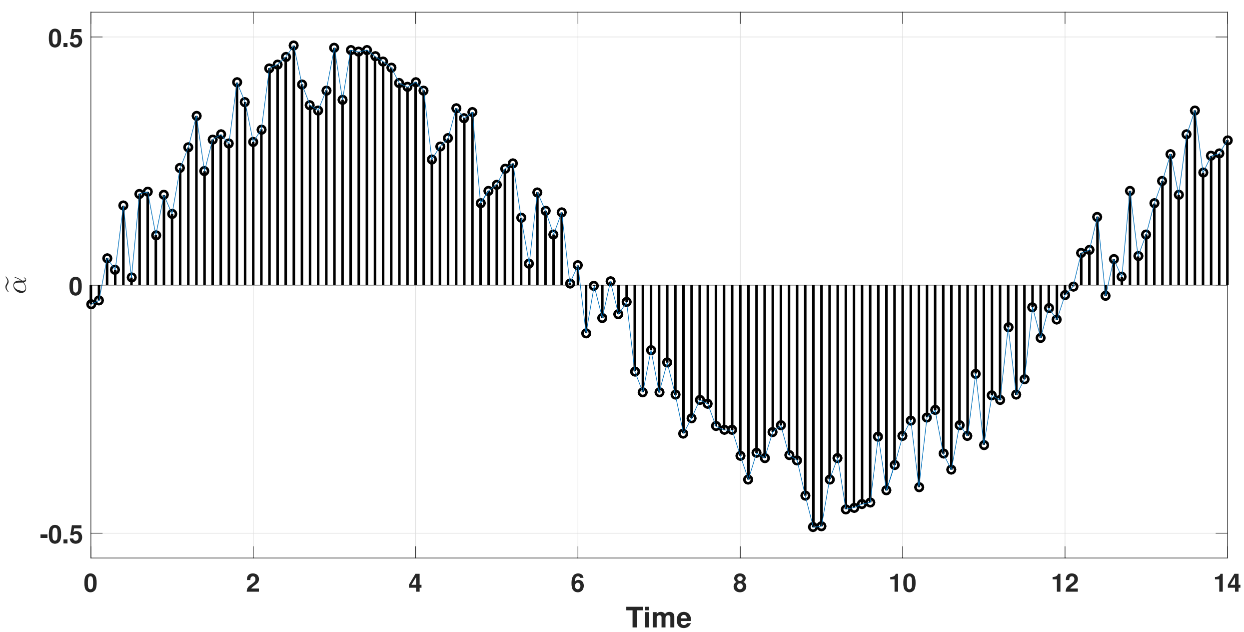
Example: The system considered is a multi-input and multi-output (MIMO) LPV system with uncertain scheduling parameters defined in equation (4.15) bounded as in equation (4.16). The system is in the form defined in equation (4.17) where

$$\begin{cases} \mathbf{A}(\widetilde{\theta}_k) = \mathbf{A}_0 + \widetilde{\theta}_1(k)\,\mathbf{A}_1; \\ \mathbf{A}_0 = \begin{bmatrix} -1 & 3 \\ 0 & -4 \end{bmatrix}; \mathbf{A}_1 = \begin{bmatrix} -2 & 5 \\ 0 & -3 \end{bmatrix}; \mathbf{B}_{u0} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; \mathbf{C}_{z0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \mathbf{C}_{y0} = \begin{bmatrix} 1 & 0 \\ 0 & 0.56 \end{bmatrix}; \end{cases} \tag{4.21}$$

$\mathbf{B}_\omega(k)$ is a diagonal matrix with proper dimensions; other matrices are zeros matrices with proper dimension; and scheduling parameter and its uncertainty bounds are $-0.5 \le \widetilde{\theta}_1(k) \le 0.5$ and $-0.1 \le \vartheta_1(k) \le 0.1$, respectively. In addition, the sensor used in simulation study is assumed



(a)



(b)

Figure 4.5: Scheduling parameter used in (4.21) with (a) 0% (b) 10% uncertainty. The x-axis is linear in time and y-axis is linear for magnitude of (a) true scheduling parameter θ (b) measured scheduling parameter $\widetilde{\theta}$.

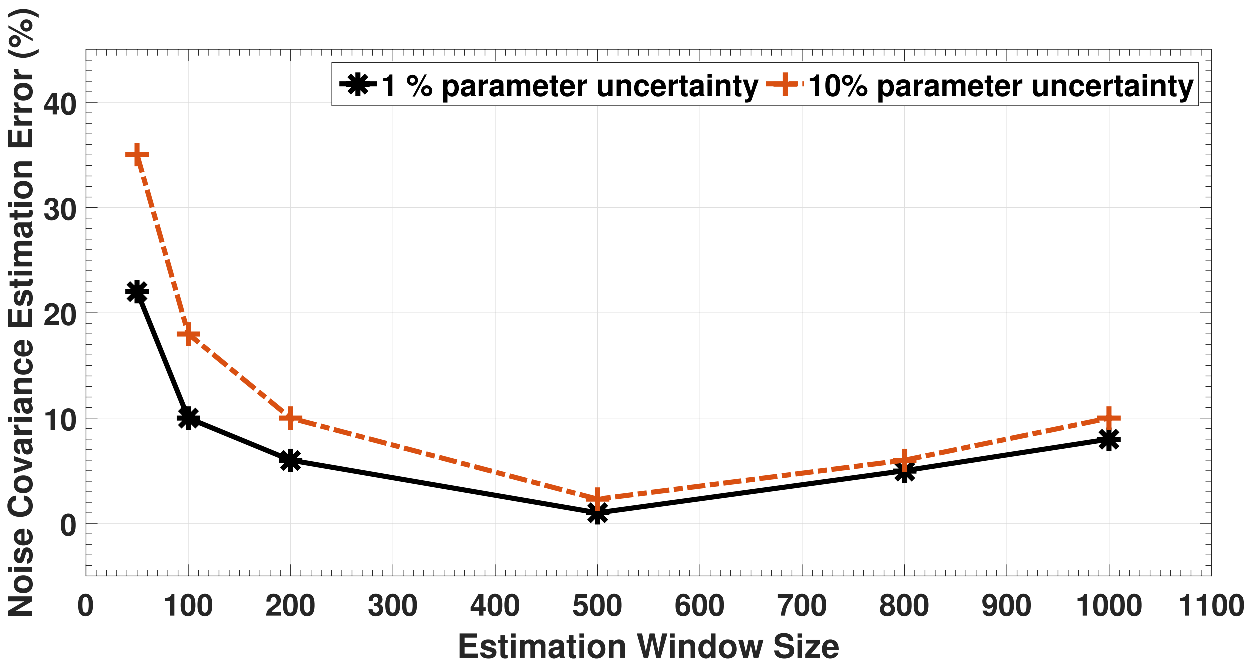
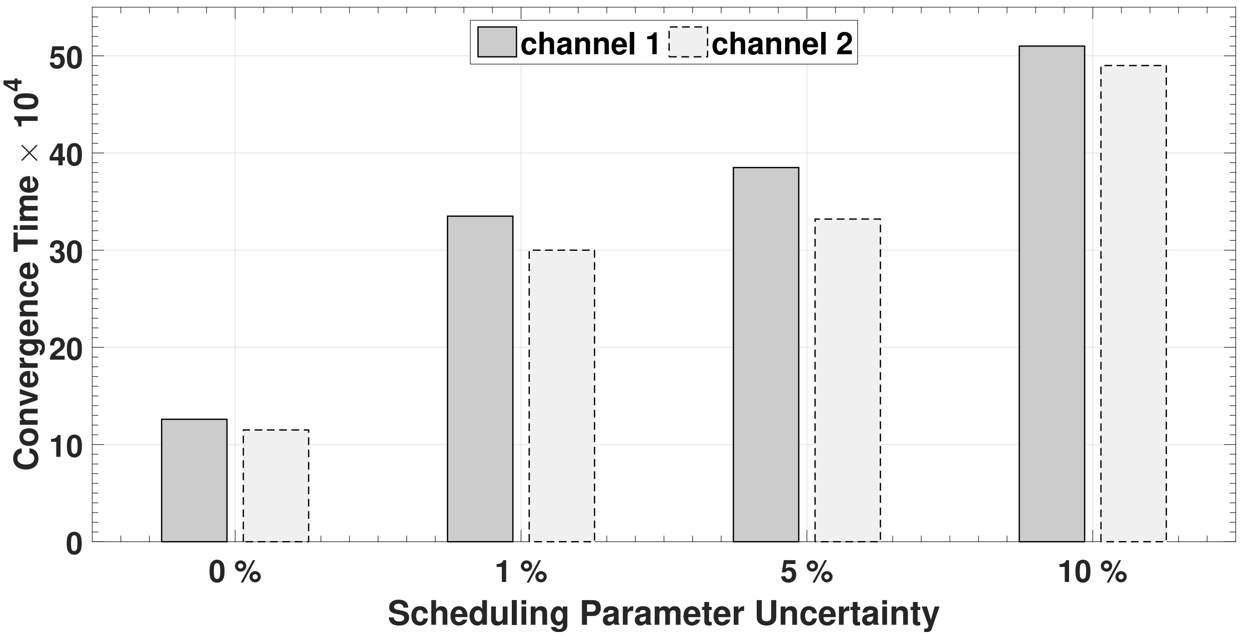
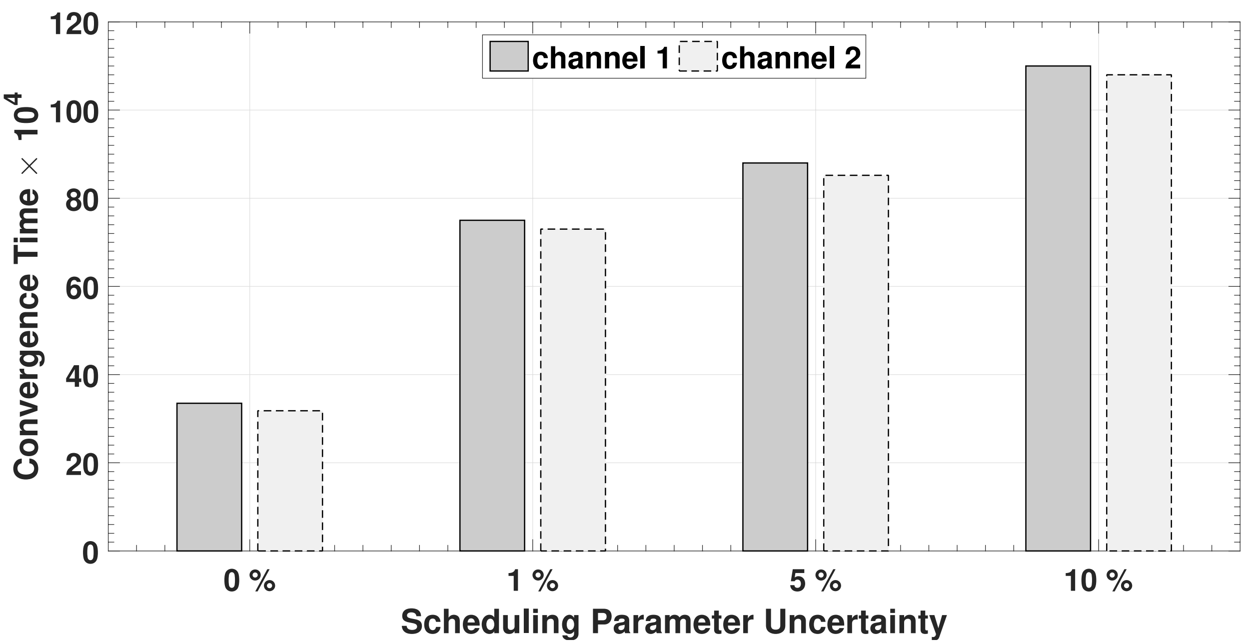


Figure 4.6: Noise covariance estimation window size and the corresponding noise covariance estimation error of system (4.21), where the noise covariance increases linearly with variation rate of $9 * 10^{-4}$ and scheduling parameter has 1% or 10% uncertainty. The noise covariance estimation error is calculated at the last time epoch of simulation run.

to have a noise covariance variation rate of $9 * 10^{-4}$ with linear or exponential increment and the sampling time is 0.1 second. The scheduling parameters used in equation (4.21) with 0% and 10% uncertainty (i.e., measurement error) are shown in Fig.4.5a and Fig. 4.5b, respectively, where the x-axis is linear in time and y-axis is linear for magnitude. Henceforth, through the simulation, the time required for the estimated noise covariance to converge to the actual one is referred as *convergence time* and the sensor noise covariance estimation error calculated at the last epoch of simulation run is referred as *noise covariance estimation error*. Simulation results in Fig. 4.6 present the noise covariance estimation window size and the corresponding noise covariance estimation errors, where channel 1 has scheduling parameter with 1% or 10% uncertainty and noise covariance increases linearly at a rate of $9*10^{-4}$. The case with estimation window size of 500 samples shows the smallest noise covariance estimation error. Therefore, the estimation window size is set to 500, and the lower and upper bounds are set to 400 and 600, respectively. Consequently, the forgetting factor is to 0.998,



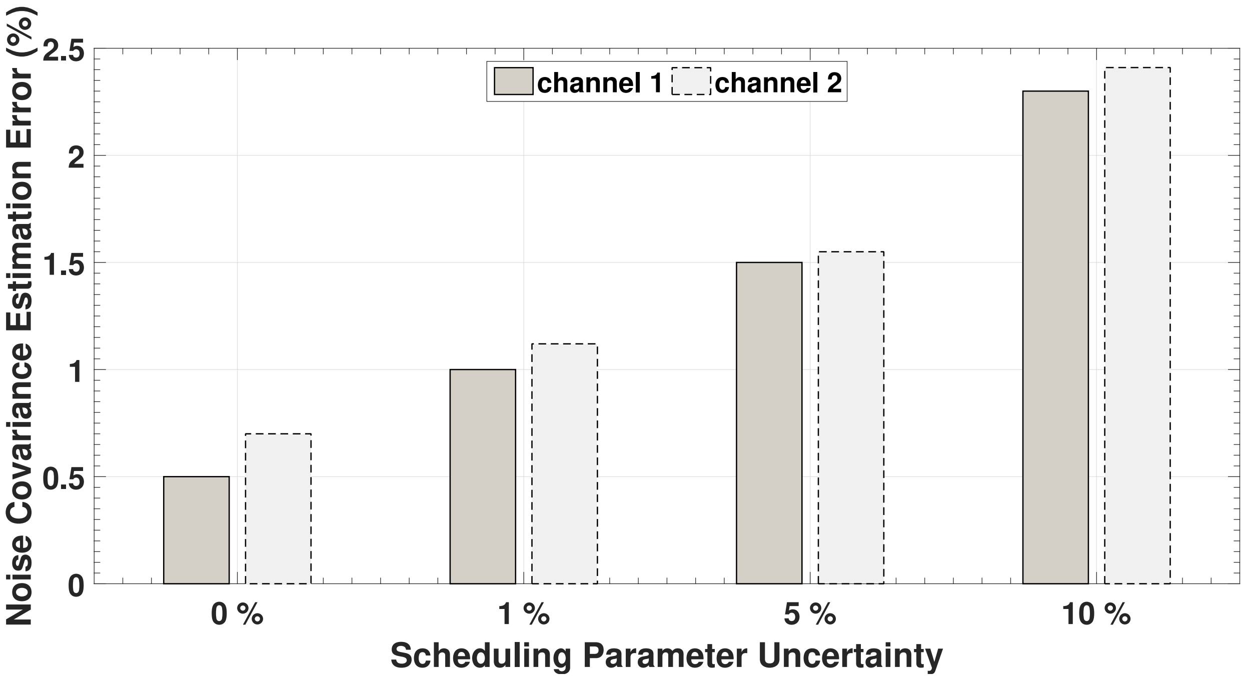
(a)



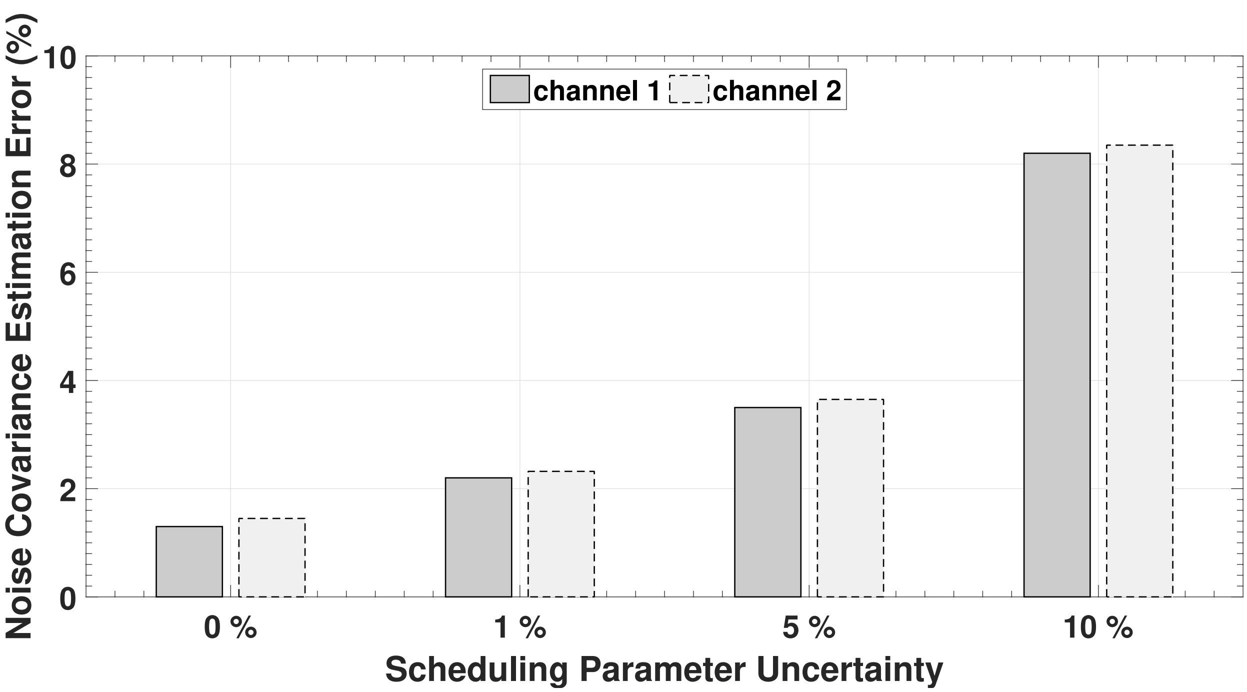
(b)

Figure 4.7: Channel 1 and channel 2 noise covariance estimation convergence time, where scheduling parameter has 0%, 1%, 5% or 10% uncertainty and the noise covariance has variation rate of $9 * 10^{-4}$ with (a) linear (b) exponential increment.

see Chapter 3 for more details. The noise covariance estimation convergence time for channels 1 and 2 is presented in Fig. 4.7, where scheduling parameter has 0%, 1%, 5% or 10% uncertainty and



(a)

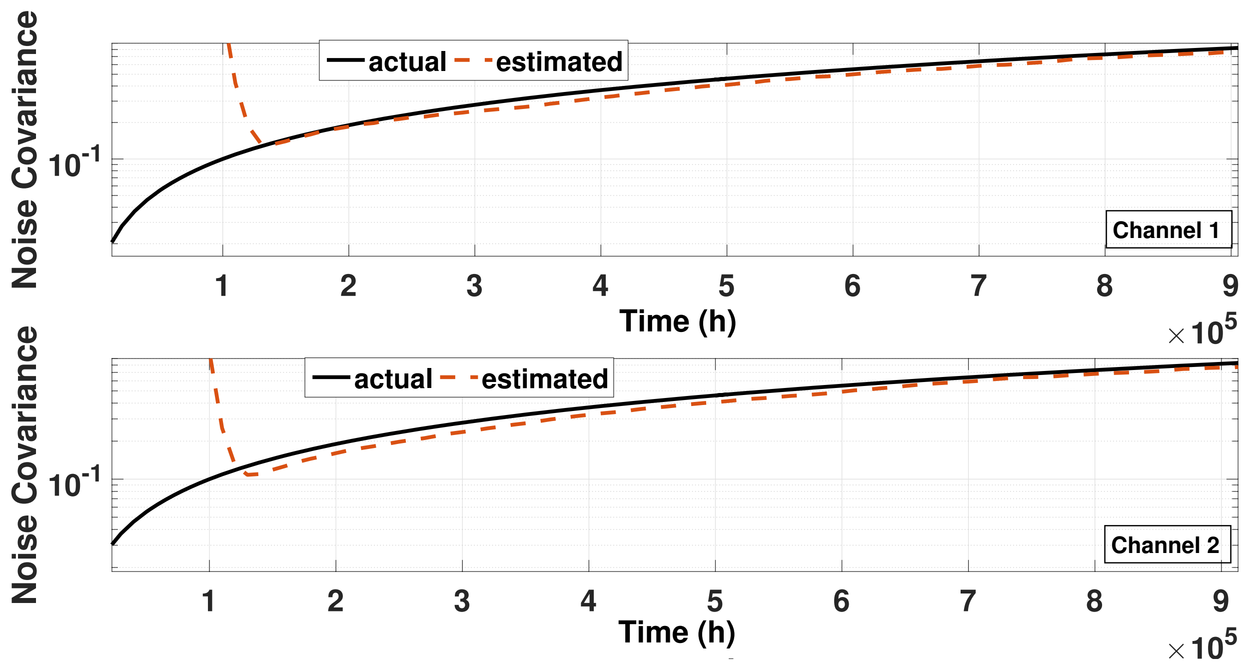


(b)

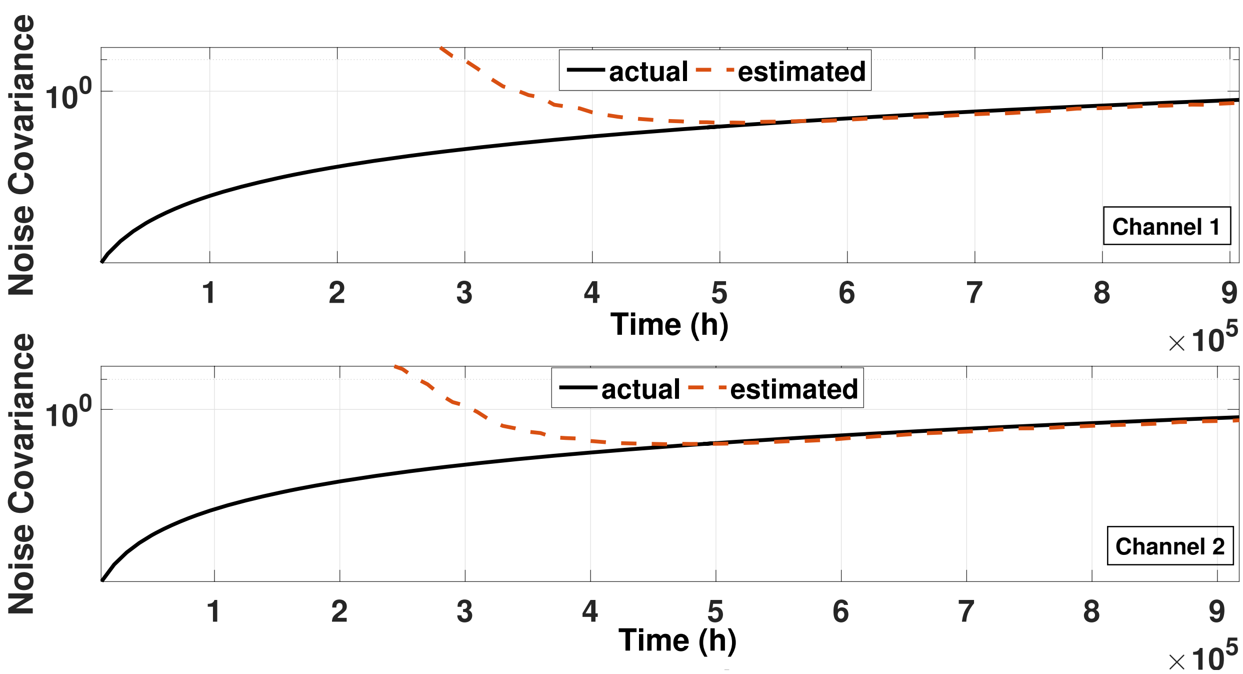
Figure 4.8: Channel 1 and channel 2 noise covariance estimation error, where scheduling parameter has 0%, 1%, 5% or 10% uncertainty and the noise covariance has variation rate of $9 * 10^{-4}$ with (a) linear (b) exponential increment.

the sensor noise covariance has linear or exponential increment. The noise covariance estimation error for channels 1 and 2 is presented in Fig. 4.8, where scheduling parameter has 0%, 1%, 5% or 10% uncertainty and the sensor noise covariance has linear or exponential increment. The results summarized in Figs. 4.7 and 4.8 show that for both system (4.21) channels as the scheduling parameter uncertainty increases, the noise covariance estimation convergence time and estimation error increase but with a reasonable tracking performance. The simulation results in Figs. 4.9a and 4.9b present the actual and estimated sensor noise covariances for both channels of LPV system (4.21) with linearly increased sensor noise covariance and scheduling parameters have 0% and 10% uncertainties, respectively. Figs. 4.10a and 4.10b show the previous simulation scenarios with exponentially increased sensor noise covariance. For Figs. 4.9a to 4.10b , the upper and lower sub-figures represent channels 1 and 2 of system (4.21), respectively. The x-axis is linear in time, and y-axis is logarithmic and represents the noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively. Through the simulation results (see Figs. 4.9a to 4.10b) the estimated sensor noise covariances show a reasonable tracking performance to the actual ones under slowly-varying noise covariance scenario. Indeed, when the scheduling parameter uncertainty increases, the noise covariance estimation algorithm needs more time to converge and the noise covariance estimation error increases. In other words, the proposed algorithm for estimating the gradual sensor performance degradation due to sensor aging converges fast under a small scheduling parameter uncertainty. The computation performance for the proposed algorithm can be indicated by the time consumed by CPU for one iteration of noise covariance estimation and it is found 0.009375 sec for the mentioned computer with a sampling time of 0.1 sec, which shows the ability for the proposed algorithm to estimate the gradual sensor performance degradation online.

On the other hand, if the sensor noise covariance abruptly (or intermittently) changes, the proposed memory-based technique for detecting the abrupt (or intermittent) noise covariance change can be

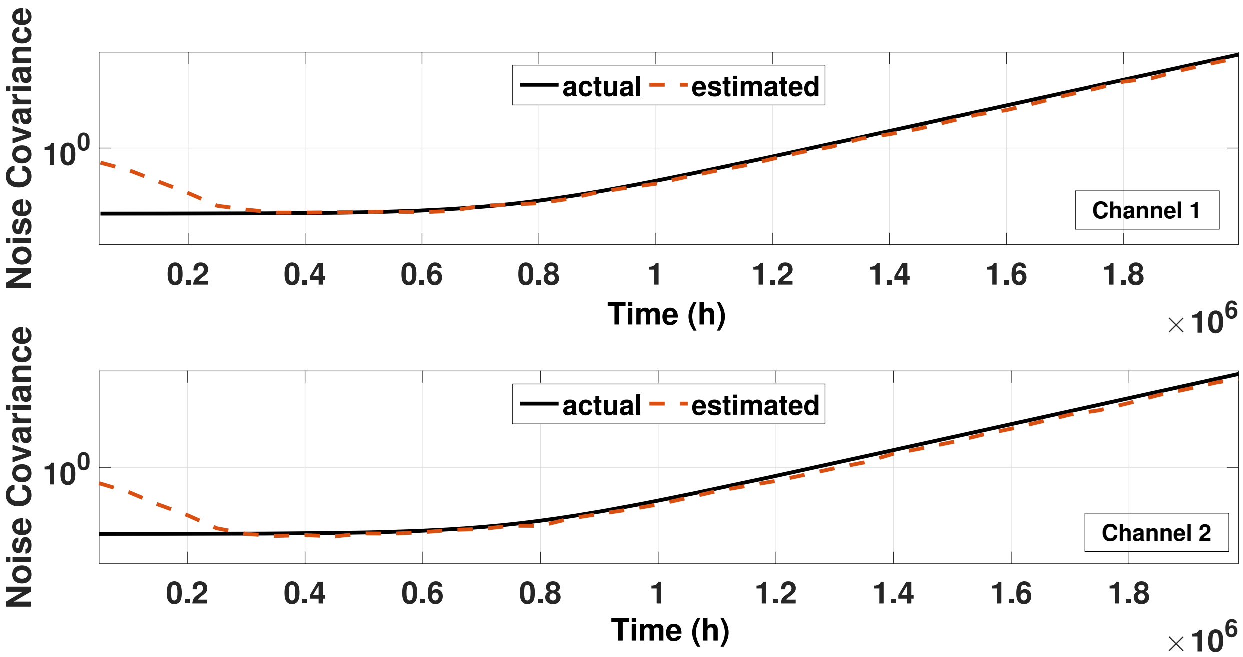


(a)

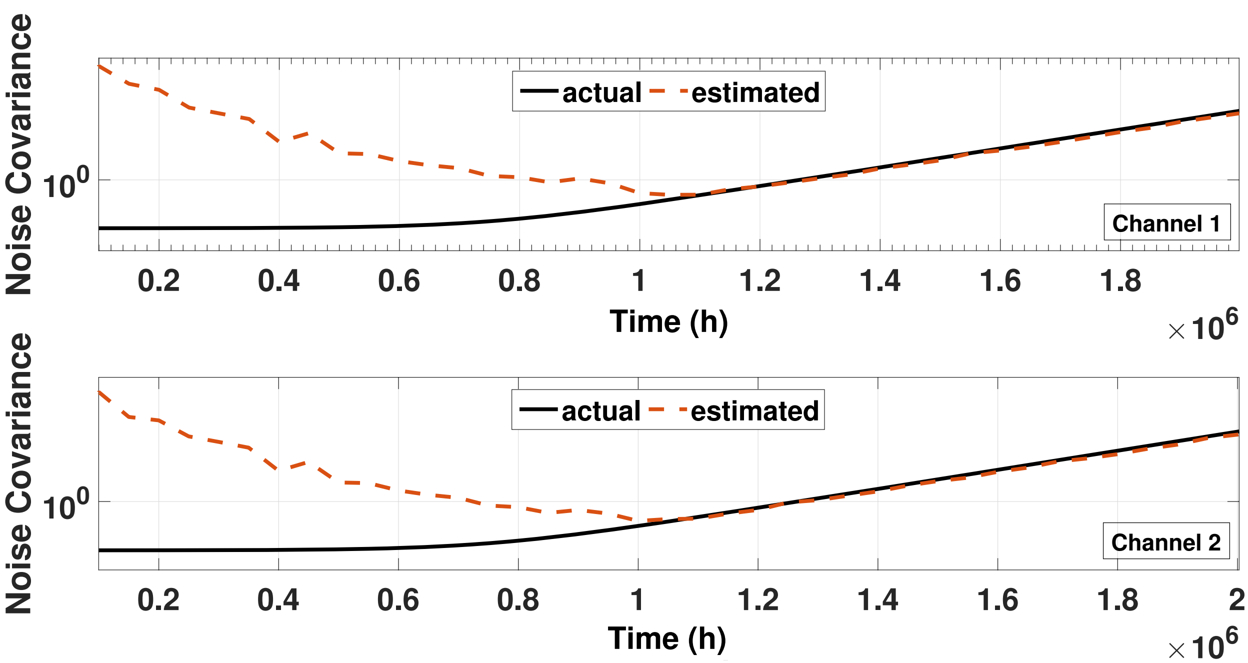


(b)

Figure 4.9: Actual and estimated sensor noise covariances matrices for the system (4.21) with linearly increased sensor noise covariance and with scheduling parameter uncertainty, where the estimation window size is set to 500. The upper and lower figures represent channels 1 and 2, respectively, where The x-axis is linear in time, and y-axis is logarithmic for noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively, where scheduling parameter has (a) 0% (b) 10% uncertainty.

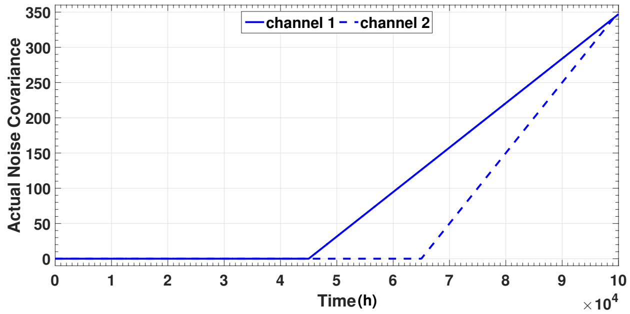


(a)

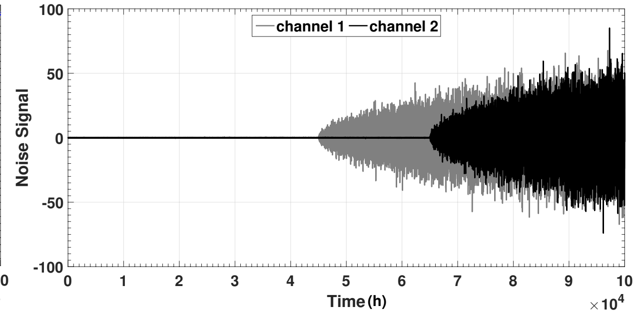


(b)

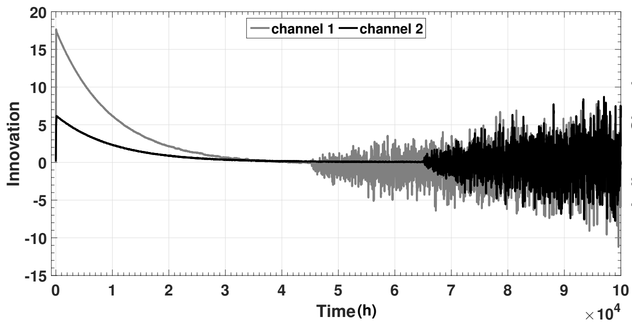
Figure 4.10: Actual and estimated sensor noise covariances matrices for the system (4.21) with exponentially increased sensor noise covariance and with scheduling parameter uncertainty, where the estimation window size is set to 500. The upper and lower figures represent channels 1 and 2, respectively. The x-axis is linear in time, and y-axis is logarithmic for noise covariance. The solid and dotted lines represent the actual and estimated noise covariance, respectively, where scheduling parameter has (a) 0% (b) 10% uncertainty.



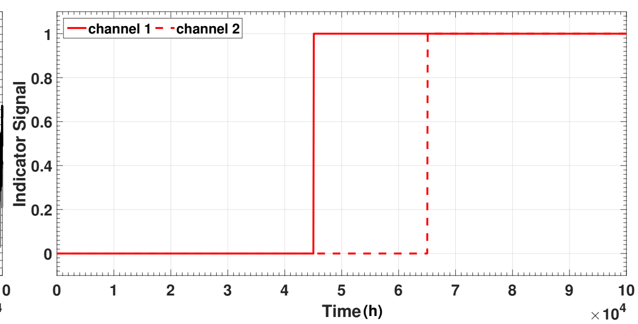
(a) Actual noise covariance.



(b) Noise signal.



(c) Innovation.



(d) Indicator signal.

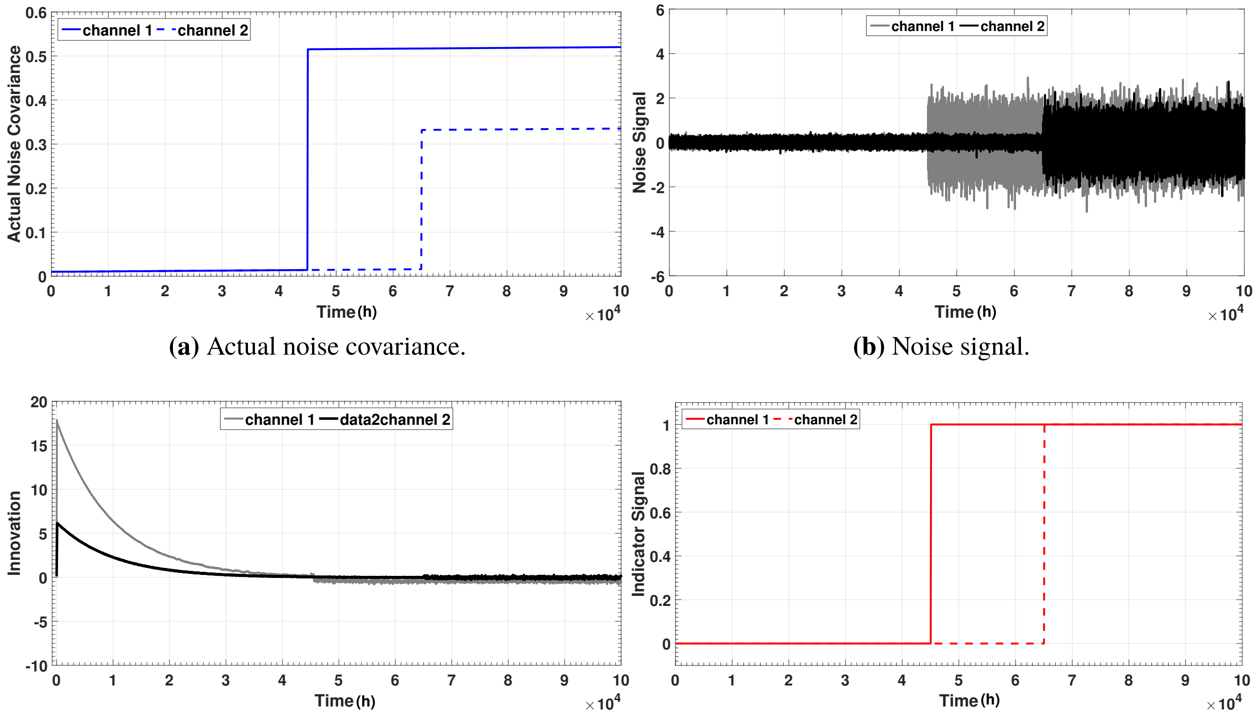
Figure 4.11: Channel 1 and channel 2 sensor measurement noise covariance with abrupt increment change at different times, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

applied. We consider different scenarios to simulate the worst cases where both sensor channels fail, as follow,

Scenario 1: both output sensor channels fail at a different time. Channel 1 output sensor fails before channel 2, see Fig. 4.11, where the sensor noise covariance increases slowly and then suddenly with a large increment (i.e., an abrupt increment of sensor noise covariance); and channel 1 sensor after change variation rate is less than that of the channel 2.

Scenario 2: both output sensor channels fail at different time with channel 1 sensor fails before that of channel 2, where the sensor noise covariance increase slowly and suddenly jump to a large value; and channel 1 sensor noise covariance is larger than that of channel 2, see Fig. 4.12.

Scenario 3: both output sensor channels fail at different time, where channel 1 sensor fails after channel 2; the sensor noise covariance has an intermittent jump for certain duration and channel 2



(a) Actual noise covariance.

(b) Noise signal.

(c) Innovation.

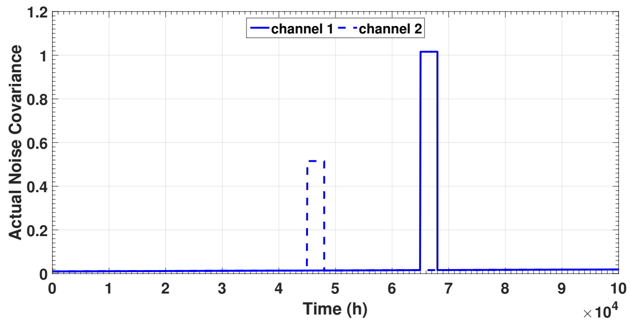
(d) Indicator signal.

Figure 4.12: Channel 1 and channel 2 sensor measurement noise covariance with abrupt jump change at different times, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

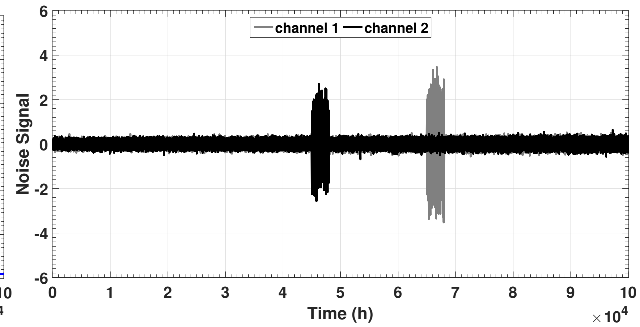
sensor noise covariance jump less than that of channel 1, see Fig. 4.13.

Scenario 4: both output sensor channels fail at different time, that is, channel 1 sensor fails after channel 2. Fig. 6.3 presents the scenario where channel 1 sensor noise covariance increases slowly, and suddenly increases with large variation rate and channel 2 sensor noise covariance has an intermittent jump, which is less than the maximum value of channel 1 sensor noise covariance, for certain duration.

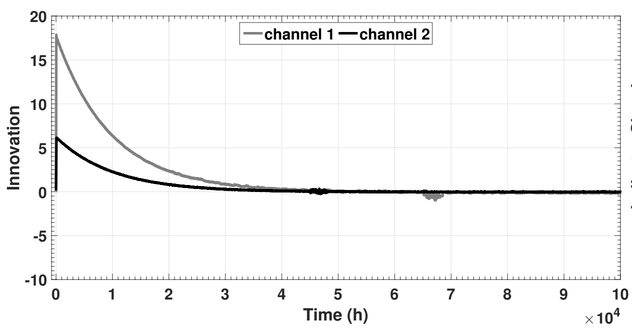
Scenario 5: both output sensor channels fail at different time, channel 1 sensor fails before channel 2, where channel 1 sensor noise covariance has an intermittent jump for certain duration and channel 2 sensor noise covariance increases slowly, and then suddenly jumps to a large value which is less than channel 2 sensor noise covariance maximum jump. Fig. 4.15 shows this scenario.



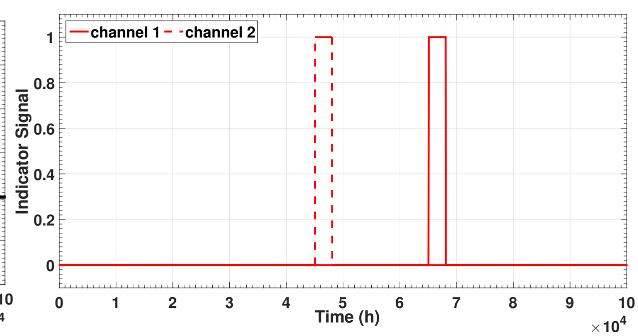
(a) Actual noise covariance.



(b) Noise signal.



(c) Innovation.

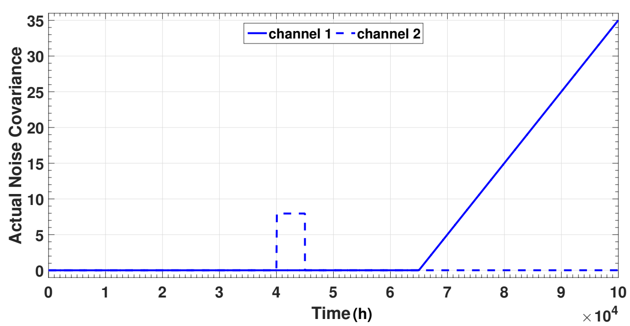


(d) Indicator signal.

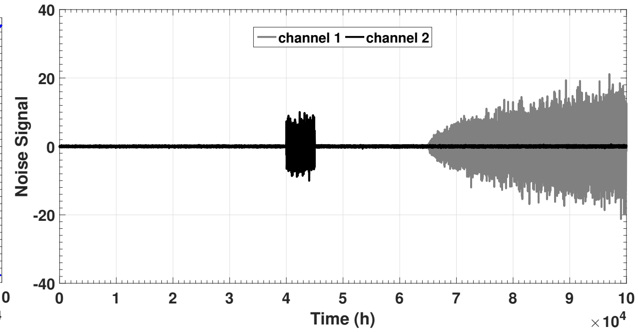
Figure 4.13: Channel 1 and channel 2 sensor measurement noise covariance with intermittent jump change for some duration at different times, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

Scenario 6: both sensor channels fail simultaneously, see Fig. 4.16, where channel 2 sensor noise covariance has an intermittent jump for certain duration while channel 1 increase slowly, and then suddenly jump to a large value which is less than channel 2 sensor noise covariance maximum jump.

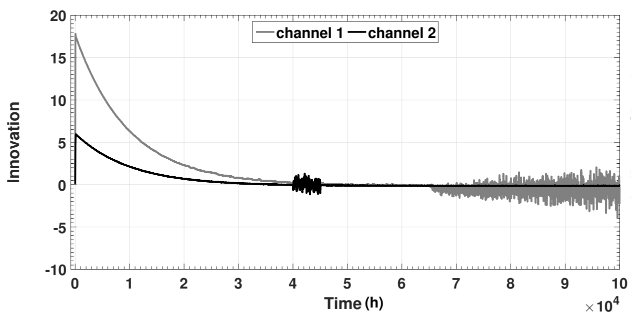
For Figs. 4.11 to 4.16; the sub-figure (a) is the actual noise covariance; the sub-figure (b) is the noise signal; (c) is the innovation signal; (d) is the sensor fault indicator signal, where the x-axis is linear in time and y-axis is linear and represents the value correspond to each sub-figure. Simulation results in Figs. 4.11 to 4.16 show the capability of the proposed change detection algorithm to detect and identify different abrupt or intermittent sensor fault scenarios with a fairly fast response. As a summary, the proposed algorithm shows its capability of online



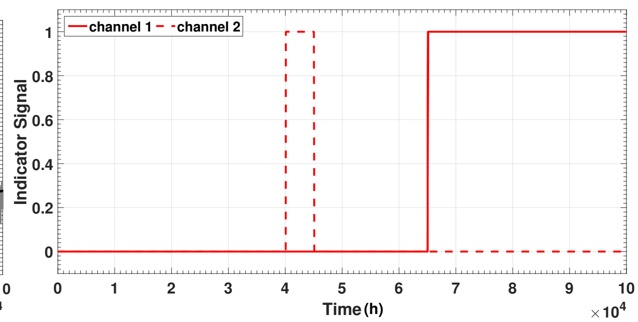
(a) Actual noise covariance.



(b) Noise signal.



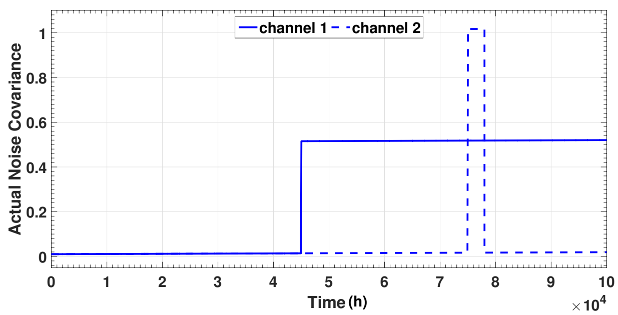
(c) Innovation.



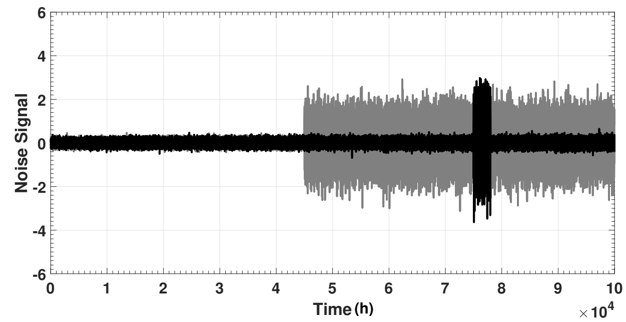
(d) Indicator signal.

Figure 4.14: Channel 1 and channel 2 sensor measurement noise covariance with channel 1 sensor noise covariance has abrupt increment jump change and channel 2 one has intermittent jump change for some duration at different time, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

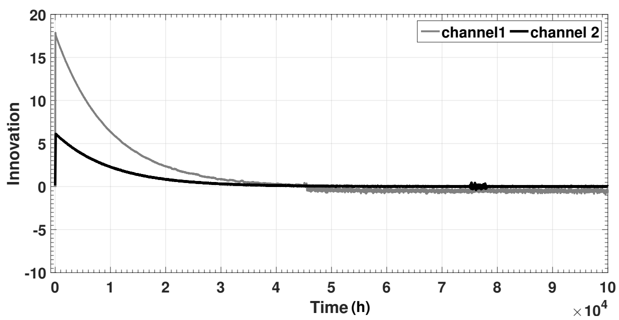
monitoring the sensor performance deterioration and detecting sensor failure. That is, it estimates the slow-varying sensor noise covariance under different covariance variation functions and different scheduling parameter uncertainties, and detects the noise covariance under different abrupt or intermittent change scenarios when the estimation window length is properly selected. In other words, the proposed algorithm is able to provide information of sensor performance regarding its noise covariance and to detect sensor fault.



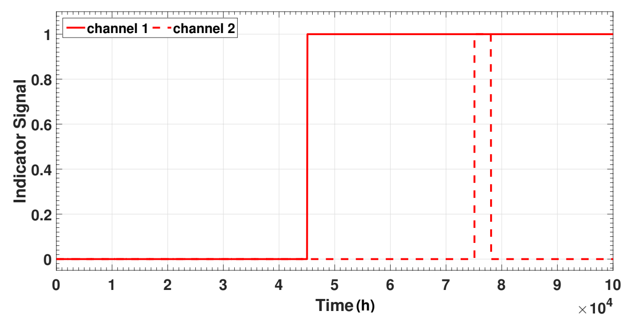
(a) Actual noise covariance.



(b) Noise signal.



(c) Innovation.

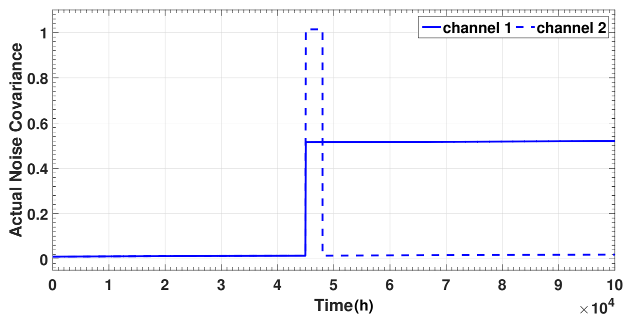


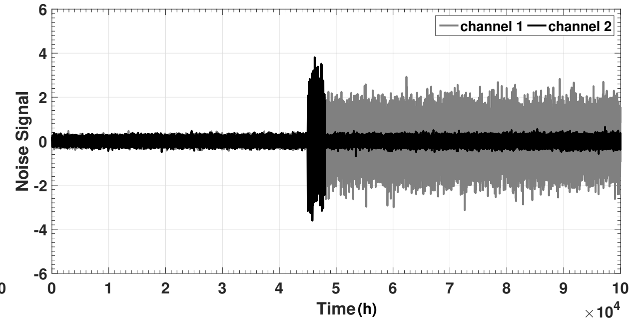
(d) Indicator signal.

Figure 4.15: Channel 1 and channel 2 sensor measurement noise covariance with channel 1 sensor noise covariance has abrupt jump change and channel 2 one has intermittent jump change for some duration at different time, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

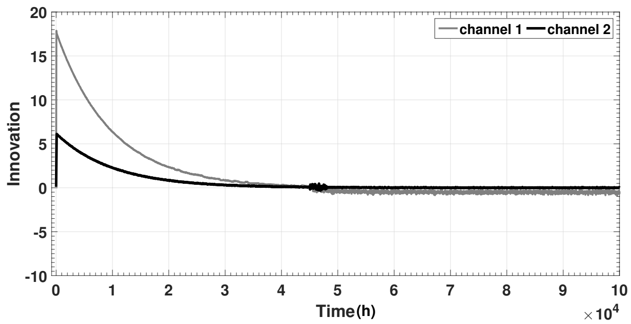
4.7 Summary

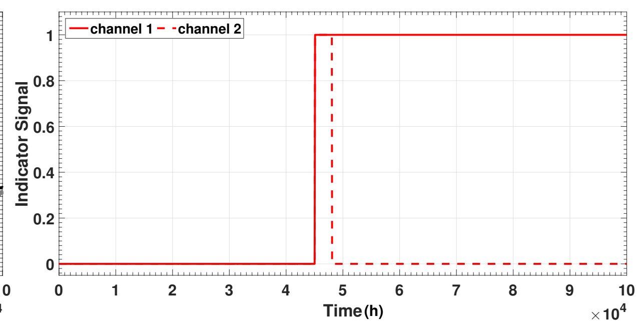
Adapting the control system to sensor performance variation helps to maintain closed-loop system performance under varying sensor performance and to avoid system catastrophic failure, performance degradation, system shut down, or fatal accident. Our work focuses on online monitoring the sensor performance deterioration and detecting sensor failure, assuming that the sensor performance shift can be characterized by the noise covariance variation of the sensor measurement. An algorithm is proposed for estimating the slowly-varying sensor measurement noise covariance due to the sensor aging and detecting the abrupt or intermittent change of sensor measurement noise covariance. The proposed algorithm is developed for LTV systems and applied to LPV systems as a special case. The iterative nature of the proposed algorithm significantly reduces the computational





(a) Actual noise covariance. **(b)** Noise signal.





(c) Innovation. **(d)** Indicator signal.

Figure 4.16: Channel 1 and channel 2 sensor measurement noise covariance with channel 1 sensor noise covariance has abrupt jump change and channel 2 one has intermittent jump change for some duration at the same time, where (a) actual noise covariance (b) noise signal (c) innovation (d) indicator signal. The x-axis is linear in time.

load and make it feasible for real-time applications. Simulation investigation demonstrates that the proposed algorithm converges when the estimation window is properly chosen.

Chapter 5

Guarantee Performance ICC-LPV Control with Sensor Aging

5.1 Introduction

Many physical systems are subject to possible component malfunctions, which may cause significant system performance degradation and even instability. The requirement for dependable and repeatable system performance is essential for safety-critical applications, where a successful mission of protecting human life, property, and/or environment becomes a paramount goal. To minimize the possibility of unexpected system failures, control systems need to take account for system reliability; see [Ram and Davim, 2018]. To improve the system reliability, sensor performance degradation information needs to be considered by the controller at each moment. To be specific, sensor failure affects the system output performance directly and could lead to catastrophic consequences [Cai and Wu, 2010].

Indeed, in practical applications, all physical control systems have to operate under certain actuator constraints since these actuators have a finite amount of available power. When closed-loop controllers are designed using the conventional methods, these actuator constraints are usually not taken into account. Consequently, it is possible to design a controller that commands more control power than its capability. As a result, some or all control inputs could saturate and overall system performance degrades. On the other hand, the input covariance constraints (ICC) control problem is an optimal control problem, where the output $\mathcal{H}_2$ performance is minimized subject to multiple constraints on the control inputs covariances ($\mathcal{H}_2$ performances). Then, with the ICC control design, the actuator constraints are considered during the controller synthesis process. Therefore, the controller with the best possible performance is obtained with respect to a given set of available actuator constraints. The ICC control problem is an optimal control problem was proposed by [Zhu et al., 1995] for Hubble telescope. Then, it was presented in many literature for different applications such as model predictive control [Kocijan et al., 2004], broadcast channel systems [Weingarten et al., 2006], wiretap channel systems [Fakoorian and Swindlehurst, 2013], multi-cell distributed antenna system [Feng et al., 2013], economic optimization [Van Hessem et al., 2001], full-duplex Gaussian relay channel [Khina et al., 2012], linear time-invariant system [White et al., 2013c], electronic throttle [Zhang et al., 2015], radio communication [Devroye et al., 2006], and cellular network [Huh et al., 2011] applications.

With this intention, the primary goal of this chapter is to design a multi-objective gain-scheduled controller, with guaranteed stability, that minimizes the output covariance cost ($\mathcal{H}_2$ norm) function subject to the multiple input covariance constraints (ICC) and $\mathcal{H}_\infty$ performance constraint in the presence of gradual sensor performance degradation due to sensor aging.

The main contributions of this chapter is a method using parametrized linear matrix inequalities (PLMI) to synthesize a multi-objective gain-scheduled noisy output-feedback controller (GSNOF), considering the sensor aging information, such that system stability and desired performances are guaranteed with optimal output covariance performance subject to constraints on the control input covariance matrix and $\mathcal{H}_\infty$ performance. Additionally, the sensor measurement noise covariance is used to model the sensor performance deterioration (fault). Gain-scheduling parameters with a combination of the time-varying parameters and the estimated sensor measurement noise covariance are called fully gain-scheduled control; gain-scheduled by the estimated sensor measurement noise covariance is called partial gain-scheduled control; and the robust controller is not gain-scheduled. Those three groups of controllers are validated and compared through simulation studies for a discrete-time linear parameter varying (LPV) system with gradually performance degraded sensors. The synthesized controller not only guarantees the closed-loop system stability but also minimizes system performance output $\mathcal{H}_2$ norm subject to constraints on the control input covariance matrix and $\mathcal{H}_\infty$ performance. The designed controller is also feasible for real-time applications; for more details see [Madhag and Zhu, 2018] or [Madhag and Zhu, a].

This chapter is organized as follows. Sub-section 5.2 provides an overview of the proposed problem, related mathematical preliminaries, and the discrete-time polytopic LPV system. Sub-section 5.3 presents the synthesize of the parametrized LMI (PLMI) conditions for the proposed mixed ICC and $\mathcal{H}_\infty$ control problem. Sub-section 5.4 provides the simulation study results. Conclusions are drawn in Sub-section 5.5.

5.2 Problem Definition

5.2.1 System Model

Consider a general discrete-time linear parameter varying (LPV) system model described in the following state-space representation

$$
\begin{aligned}
\mathrm{x}(k+1) &= \mathbf{A}(\theta(k))\mathrm{x}(k) + \mathbf{B}_u(\theta(k))\mathrm{u}(k) + \mathbf{B}_{\infty\omega}(\theta(k))\omega_\infty(k) + \mathbf{B}_{2\omega}(\theta(k))\omega_2(k), \\
\mathrm{z}_\infty(k) &= \mathbf{C}_{\infty z}(\theta(k))\mathrm{x}(k) + \mathbf{D}_{\infty u}(\theta(k))\mathrm{u}(k) + \mathbf{D}_{\infty\omega}(\theta(k))\omega_\infty(k), \\
\mathrm{z}_2(k) &= \mathbf{C}_{2z}(\theta(k))\mathrm{x}(k) + \mathbf{D}_{2u}(\theta(k))\mathrm{u}(k), \\
\mathrm{y}(k) &= \mathbf{C}_y\,\mathrm{x}(k) + \widetilde{\nu}(k)
\end{aligned}
\tag{5.1}
$$

where $k \in \mathbb{Z}_+$ is the sample time index; $\mathrm{x}(k) \in \mathbb{R}^n$ is the state vector; $\mathrm{z}_2(k) \in \mathbb{R}^{p_2}$ is the system performance output vector; $\mathrm{z}_\infty(k) \in \mathbb{R}^{p_\infty}$is the controlled output vector related to modeling error; $\mathrm{y}(k) \in \mathbb{R}^q$ is the measured output vector; $\mathrm{u}(k) \in \mathbb{R}^m$ is the control input vector; $\mathbf{A}(\theta(k)) \in \mathbb{R}^{n\times n}$; $\mathbf{B}_u(\theta(k)) \in \mathbb{R}^{n\times m}$; $\mathbf{B}_{\infty\omega}(\theta(k)) \in \mathbb{R}^{n\times r_{\infty\omega}}$; $\mathbf{B}_{2\omega}(\theta(k)) \in \mathbb{R}^{n\times r_{2\omega}}$; $\mathbf{C}_{2z}(\theta(k)) \in \mathbb{R}^{p_2\times n}$; $\mathbf{C}_{\infty z}(\theta(k)) \in \mathbb{R}^{p_\infty\times n}$; $\mathbf{C}_y \in \mathbb{R}^{q\times n}$; $\mathbf{D}_{\infty u}(\theta(k)) \in \mathbb{R}^{p_\infty\times m}$; $\mathbf{D}_{\infty\omega}(\theta(k)) \in \mathbb{R}^{p_\infty\times r_{\infty\omega}}$; $\mathbf{D}_{2u}(\theta(k)) \in \mathbb{R}^{p_2\times m}$; and the scheduling parameters vector $\theta(k)$ is defined in Eqn. (5.3). The parameter-dependent matrices for system (5.1) are assumed to be affine with respect to the time-varying scheduling parameters vector. The initial state vector x(0) in system (5.1) is assumed to have a normal distribution with zero mean ($\mathbb{E}[\mathrm{x}(0)] = 0$) and covariance ($\mathbf{P}_0 = \mathbb{E}(\mathrm{x}(0)\mathrm{x}^T(0)) > 0$). System (5.1) is assumed to be observable and controllable. Note that, the observability and controllability assumptions are used mainly due to the fact that the observability and controllability for LPV systems are not well defined in the LPV control literature. The terms $\omega_\infty(k) \in \mathbb{R}^{r_{\infty\omega}}$, $\omega_2(k) \in \mathbb{R}^{r_{2\omega}}$ and $\widetilde{\nu}(k) \in \mathbb{R}^{r_{\widetilde{\nu}}}$ in system (5.1) denote the disturbance (noise) inputs due to modeling error, process and measurement noises respectively. In addition, they are assumed to be uncorrelated Gaussian white ones with their means and covariances defined by

$$\begin{cases} \mathbb{E}\left[\omega_2(k)\right] = 0, \;\; \mathbb{E}\left[\omega_2(k)\omega_2^T(j)\right] = \mathbf{W}_{k_2}\,\delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \\ \mathbb{E}\left[\omega_\infty(k)\right] = 0, \;\; \mathbb{E}\left[\omega_\infty(k)\omega_\infty^T(j)\right] = \mathbf{W}_{k_\infty}\,\delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \\ \mathbb{E}\left[\widetilde{\nu}(k)\right] = 0, \;\; \mathbb{E}\left[\widetilde{\nu}(k)\widetilde{\nu}^T(j)\right] = \mathbf{I}\,\delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \\ \mathbb{E}\left[\omega_2(k)\widetilde{\nu}^T(j)\right] = \mathbb{E}\left[\omega_\infty(k)\widetilde{\nu}^T(j)\right] = \mathbb{E}\left[\omega_2(k)\mathrm{x}^T(j)\right] = \mathbb{E}\left[\widetilde{\nu}(k)\mathrm{x}^T(j)\right] = 0, \end{cases} \tag{5.2}$$

where δ is a Kronecker Delta function $\big($i.e., $\delta(0) = 1$ and $\delta(k) = 0; \forall\, k \neq 0\big)$. Note that $\widetilde{\nu}(k) = \mathbf{V}_k^{\frac{-1}{2}}\,\nu(k),\; \mathbb{E}\left[\nu(k)\right] = 0,\; \mathbb{E}\left[\nu(k)\nu_j^T\right] = \mathbf{V}_k\,\delta(t_k - t_j), \forall\, k, j \in \mathbb{Z}_+$. The disturbance and process noise covariance matrices, $\mathbf{W}_{k_\infty}$ and $\mathbf{W}_{k_2}$, are assumed to be positive semi-definite and known at each time instant, where the later could be assumed as an identity matrix with no conservativeness. Moreover, all sensor measurement noises are assumed to be independent (i.e., measurement noise for each channel is independent of the rest of them), that is, $\mathbf{V}_k = diag\big(\,\sigma_1(k), \cdots, \sigma_q(k)\,\big)$, where $\mathbf{V}_k$ is the measurement noise covariance matrix; and $\sigma_i(k)$ is the sensor measurement noise variance of the i^{th} channel. The scheduling parameters vector is defined as

$$\theta(k) = \left[\,\theta_1(k), \cdots, \theta_q(k),\; \theta_{q+1}(k), \cdots, \theta_{n_\theta}(k)\,\right]^T, \tag{5.3}$$

where parameters $\theta_1(k), \cdots, \theta_q(k)$ are due to sensors measurements noises covariance variation; $\theta_{q+1}(k), \cdots, \theta_{n_\theta}(k)$ are due to system parameter variations; and q is the number of measured output sensors. In addition, the parameters $\theta(k) = \left[\theta_1(k), \cdots, \theta_{n_\theta}(k)\right]^T$ are assumed to lie in a known hyper-rectangle Ω_{n_θ}; the vertices set of Ω_{n_θ} is $ver(\Omega_{n_\theta}) = \{\eta_1, \cdots, \eta_{n_\theta} |\; \eta_i \in \{\underline{\theta}_i, \overline{\theta}_i\}\}$. The scheduling parameters are assumed to be independent and time-varying parameters with the following known bounds

$$\underline{\theta}_i \leq \theta_i(k) \leq \overline{\theta}_i, \qquad \forall\, i = 1, \cdots, n_\theta \quad \forall\, k \in \mathbb{Z}_+, \tag{5.4}$$

where $\underline{\theta}_i$ and $\overline{\theta}_i$ are the lower and upper bounds of the i^{th} scheduling parameter, respectively. In addition, $\forall\ k \geq 0$, the rate of variation of each scheduling parameter is given by

$$\Delta_{\theta_i}(k) = \theta_i(k+1) - \theta_i(k), \qquad \forall\, i = 1, \cdots, n_\theta, \tag{5.5}$$

and the i^{th} bound of the scheduling parameter variation rate is

$$\underline{\Delta}_{\theta_i} \leq \Delta_{\theta_i}(k) \leq \overline{\Delta}_{\theta_i}, \quad \forall\, i = 1, \cdots, n_\theta,\ \forall\, k \in \mathbb{Z}_+, \tag{5.6}$$

where $\overline{\Delta}_{\theta_i}$ and $\underline{\Delta}_{\theta_i}$ are the lower and upper bounds of the rate of variation of the i^{th} scheduling parameter, respectively. Without loss of generality, bounds in Eqns. (5.4) and (5.6) are assumed to be symmetric.

The goal of this work is to design a multi-objective noisy static output-feedback gain-scheduling controller that exponentially stabilizes the closed-loop system and minimizes the performance output covariance subject to constraints on the control input covariance matrix and the worst case norm from input disturbances (noises) to the controlled output vector related to modeling error. To cope with the above goal, gain-scheduled methods provided by most of the relevant literature were derived for discrete-time polytopic time-varying parameter-dependent systems; see [De Caigny et al., 2010], [Lacerda et al., 2016] and [White et al., 2013c]. Therefore, in the next sub-section, the state-space representation of the considered system will be transformed into polytopic fashion so that the controller can be synthesized using existing methods. Note that, some of the notations and steps presented in the next section are borrowed from [De Caigny et al., 2010].

5.2.2 Polytopic Parameter Representation

The objective of this subsection is to present briefly the change of variable used to construct a new convex parameter space (i.e., polytopic state-space representation of the affine one). It is advised to see [Lacerda et al., 2016] and therein references for more details. The affine parameter-dependent LPV system can be represented in convex fashion by the change of variable to utilize the benefit of polytopic representation; see Fig. 5.1. Considering the scheduling parameters and its variations in Eqn. (5.5), the scheduling parameters can be mapped (i.e., $\theta_i \mapsto \alpha_i$) as follows [Lacerda et al., 2016]

$$\begin{cases} \alpha_{i1}(k) = \dfrac{\theta_i(k) + \overline{\theta}_i}{2\,\overline{\theta}_i} \\ \because\ \alpha_{i1}(k) + \alpha_{i2}(k) = 1 \quad \Longrightarrow \quad \alpha_{i2}(k) = \dfrac{\overline{\theta}_i - \theta_i(k)}{2\,\overline{\theta}_i}, \\ \therefore\ \alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\theta}(k)\right]^T, \quad \alpha_i(k) = \big(\alpha_{i1}(k), \alpha_{i2}(k)\big), \quad \forall i = 1, \cdots, n_\theta, \end{cases} \tag{5.7}$$

where $\alpha_i(k)$ is the i^{th} component of the parameters vector $\alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\theta}(k)\right]^T$ in the polytopic space. Note that, since all scheduling parameters are assumed to be bounded from above and below (see Eqn. (5.4)), the resulting space for $\alpha_i(k)$ have two vertices, that is, $\alpha_i(k) = (\alpha_{i1}(k), \alpha_{i2}(k)) \in \Lambda_2,\ \forall i = 1, \cdots, n_\theta$. Therefore, using this change of variables (i.e., Eqn. (5.7)), the affine parameter-dependent system as well as the gain-scheduling controller can be expressed in terms of scheduling parameters vector $\alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\theta}(k)\right]^T \in \Lambda_{2n_\theta}$ belong to a convex domain (i.e., polytopic space Λ_{2n_θ}). Considering the variation rate in Eqn. (5.5) and apply it to the

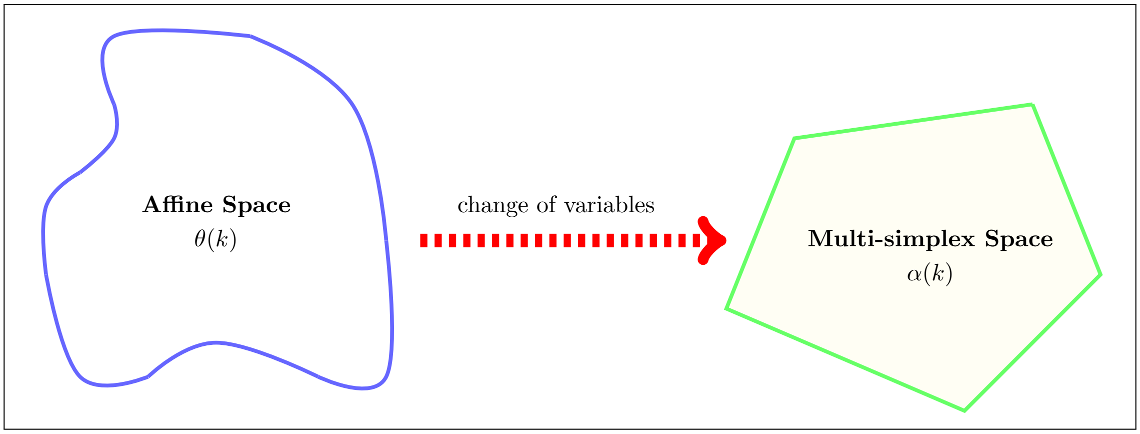


Figure 5.1: Affine space parameter to multi-simplex space mapping.

change of variable in Eqn. (5.7) yields

$$\begin{aligned}\Delta_{\alpha_{i1}}(k) &= \alpha_{i1}(k+1) - \alpha_{i1}(k),\\ &= \frac{\theta_i(k+1)+\overline{\theta}_i}{2\,\overline{\theta}_i} - \frac{\theta_i(k)+\overline{\theta}_i}{2\,\overline{\theta}_i} = \frac{\Delta_{\theta_i}(k)}{2\,\overline{\theta}_i}\end{aligned} \tag{5.8}$$

Considering $\alpha_i(k) = (\alpha_{i1}(k), \alpha_{i2}(k)) \in \Lambda_2$, it is clear that

$$\alpha_{i1}(k) + \alpha_{i2}(k) = 1, \quad \Longrightarrow \quad \Delta_{\alpha_{i2}}(k) = -\Delta_{\alpha_{i1}}(k). \tag{5.9}$$

where $\alpha_i(k) = (\alpha_{i1}(k), \alpha_{i2}(k)) \in \Lambda_2$, $\forall i = 1, \cdots, n_\theta$, is the i^{th} component of the polytopic space parameters vector $\alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\theta}(k)\right]^T \in \Lambda_{2n_\theta}$. Then, the parameters variation rates vector is $\Delta_\alpha(k) = \left[\Delta_{\alpha_1}(k), \cdots, \Delta_{\alpha_{n_\theta}}(k)\right]^T \in \Lambda_{2n_\theta}$, where $\Delta_{\alpha_i}(k) = \left(\Delta_{\alpha_{i1}}(k), \Delta_{\alpha_{i2}}(k)\right) \in \Lambda_2$, $\forall i = 1, \cdots, n_\theta$. Generally, each scheduling parameter $\alpha_{ij}(k)$ $(i = 1, \cdots, n_\theta,\ j \in [1,2])$ belongs to Λ_2, it yields

$$\sum_{i=1}^{n_\theta} \Delta_{\alpha_{ij}} = 0,\ \forall\, j \in [1,2]. \tag{5.10}$$

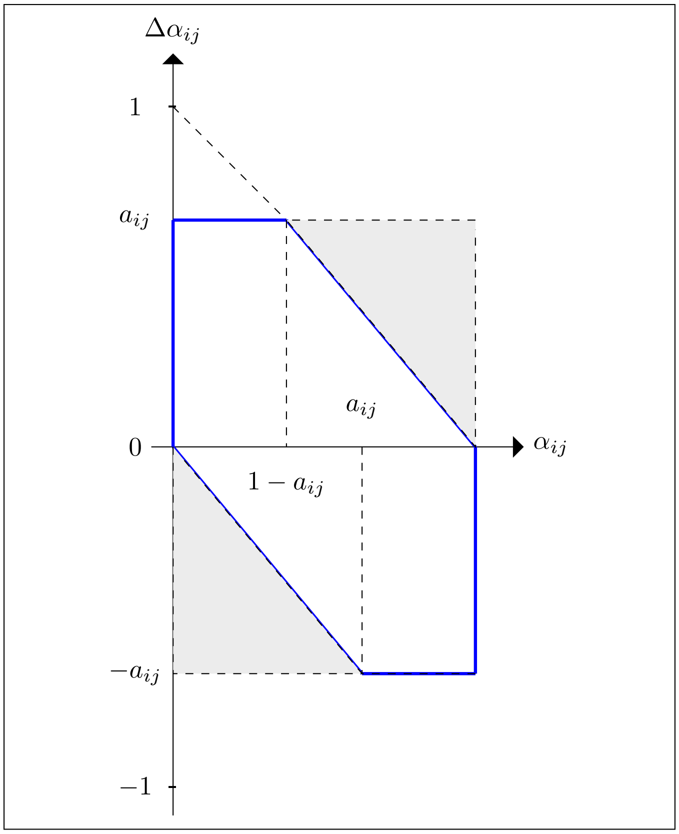


Figure 5.2: $(\alpha_{ij}, \Delta\alpha_{ij})$-space, where the area can $\Delta\alpha_{ij}$ can assume values as function of α_{ij} is the one with blue boundaries, and for more details; see [Lacerda et al., 2016] and [Oliveira and Peres, 2009].

Consider Eqns. (5.6) and (5.10), the parameter vectors α_{ij} $(i = 1, \cdots, n_\theta,\ j \in [1, 2])$ have variation rates bounds as

$$\left|\Delta_{\alpha_{ij}}\right| \leq a_{ij}, \qquad a_{ij} \in \{0, 1\}, \qquad \forall j \in [1, 2],\ \forall i = 1, \cdots, n_\theta. \tag{5.11}$$

Eqn. (5.11) represents the case where the scheduling parameters variate within a bounded rate of variation.

In addition, in the discrete-time case, the rate of variation depends on the associated value of the parameter, which is the main difference from the continuous-time case, and assuming parameter rate independent could be very conservative; see Fig. 5.2, and for more details; see [Lacerda et al., 2016], [De Caigny et al., 2010] and [Oliveira and Peres, 2009].

The uncertainty domain, where the vector $\left[\alpha^T(k), \Delta_\alpha^T(k)\right]^T \in \Lambda_{4n_\theta}$ takes values, can be modeled by the compact set below [De Caigny et al., 2010]

$$\Psi_a = \Bigg\{\psi \in \mathbb{R}^{4n_\theta} : \psi \in co\{\,\mu^1, \cdots, \mu^M\,\}, \quad \mu^j = \begin{bmatrix} f^j \\ h^j \end{bmatrix}, \quad f^j \in \mathbb{R}^{2n_\theta}, \quad h^j \in \mathbb{R}^{2n_\theta},$$
$$\sum_{i=1}^{2n_\theta} f_i^j = 1, \quad f_i^j \geq 0, \quad i = 1, \cdots, n_\theta, \qquad \sum_{i=1}^{2n_\theta} h_i^j = 0, \quad j = 1, \cdots, M\Bigg\}, \tag{5.12}$$

as a convex combination of vectors μ^j for $j = 1, \cdots, M$, where vectors f_i^j and h_i^j are given a priori; and M is the numbers of vectors μ^j, which is given by [De Caigny et al., 2010], [White et al., 2013c]

$$M = N^2 + (N-1)^2 + (N-1), \tag{5.13}$$

where N is the number of the vertices of the polytopic system (i.e., number of the scheduling parameters). For more details about Eqn. (5.13) see [De Caigny et al., 2010], [White et al., 2013c] and therein references.

This definition of the set Ψ_a ensures that $\alpha(k)$ satisfies Eqn. (5.10) for all $k \geq 0$. In addition, considering the uncertainty set Ψ_a in (5.12), each $\alpha_i(k) = (\alpha_{i1}(k), \alpha_{i2}(k)) \in \Lambda_2$ and $\Delta_{\alpha_i}(k) = (\Delta_{\alpha_{i1}}(k), \Delta_{\alpha_{i2}}(k)) \in \Lambda_2$ are given by

$$\alpha_i(k) = \sum_{j=1}^{M} f_{i\ell}^j\, \widetilde{\alpha}_j(k), \qquad \Delta_{\alpha_i}(k) = \sum_{j=1}^{M} h_{i\ell}^j\, \widetilde{\alpha}_j(k), \qquad i = 1, \cdots, n_\theta, \quad \ell = \{1, 2\}, \tag{5.14}$$

where M (which is given by Eqn. (5.13)) is the numbers of vectors μ^j (which is given by Eqn. (5.12)); $f_i^j = \left[{f_{i1}^j}^T\, {f_{i2}^j}^T\right]^T$; $h_i^j = \left[{h_{i1}^j}^T\, {h_{i2}^j}^T\right]^T$; and $\widetilde{\alpha}_j(k)$ is the j^{th} component of the scheduling parameter vector $\widetilde{\alpha}(k) = \left[\widetilde{\alpha}_1(k), \cdots, \widetilde{\alpha}_M(k)\right]^T$ in the uncertainty set Ψ_a, that is defined in Eqn. (5.12). For details about Eqn. (5.14), it is advised to see [De Caigny et al., 2010] and therein references. Note that, the representation in Eqn. (5.14) is useful to deal with parameter-dependent matrices dependent on step-ahead scheduling parameters, that is $\alpha(k+1)$, as shown later.

Next, system and controller parameter-dependent matrices with the new representation are presented. Let $\mathbf{X}(\theta(k))$ represents any parameter-dependent matrix with an affine parametrization as follows

$$\mathbf{X}(\theta(k)) = \mathbf{X}_0 + \sum_{i=1}^{n_\theta} \theta_i(k)\mathbf{X}_i, \tag{5.15}$$

where $\mathbf{X}_0 \in \mathbb{R}^{n\times n}$ and $\mathbf{X}_i \in \mathbb{R}^{n\times n}$ are constant matrices; and $\theta_i(k)$ is the i^{th} component of the scheduling parameter vector in Eqn. (5.3) at time instant k. Consider the change of variable in Eqn. (5.7), variation rate in Eqn. (5.8), and the bounds in Eqn. (5.11), then Eqn. (5.15) can be re-written as follows

$$\mathbf{X}(\alpha(k)) = \sum_{i=1}^{n_\theta} 2\,\overline{\theta}_i\,\alpha_{i1}(k)\mathbf{X}_i + \sum_{i=1}^{n_\theta} \big(\mathbf{X}_0 - \overline{\theta}_i\mathbf{X}_i\big). \tag{5.16}$$

Applying homogenization procedure that developed in [Oliveira and Peres, 2009] results

$$\mathbf{X}(\alpha(k)) = \sum_{i=1}^{n_\theta} \Big(\alpha_{i1}(k)\,\mathbf{X}_{i1} + \alpha_{i2}(k)\,\mathbf{X}_{i2}\Big), \tag{5.17}$$

where $\mathbf{X}_{i1} = \mathbf{X}_0 + \overline{\theta}_i\mathbf{X}_i$ and $\mathbf{X}_{i2} = \mathbf{X}_0 - \overline{\theta}_i\mathbf{X}_i$. Note that the homogenization steps to get (5.17) are not presented due to the space limitations; see [Oliveira and Peres, 2009] for more detail. In addition, $\mathbf{X}(\theta(k))$ represents any parameter-dependent matrix with affine parametrization while $\mathbf{X}(\alpha(k))$ represents any parameter-dependent matrix with polytopic parameter-dependent. Then, the parameter-dependent matrix is represented with a polytopic space parameters vector $\alpha(k) \in \Lambda_{2n_\theta}$ (i.e., convex domain). Next, the synthesis conditions for the proposed control problem are presented.

5.3 Control Synthesize LMIs

This section provides the synthesis parametrized LMI (PLMI) conditions for the proposed mixed ICC/$\mathcal{H}_\infty$ control problem. To make the problem tractable, the upper bound of $\mathcal{H}_2$ performances are minimized instead, subject to constraints on the $\mathcal{H}_\infty$ performance and the control input covariance matrix. Consider system (5.1) after change of variables using Eqn. (5.7), and the resulting polytopic discrete-time LPV system is given by

$$\begin{cases} \mathrm{x}(k+1) = \mathbf{A}(\alpha(k))\mathrm{x}(k) + \mathbf{B}_u(\alpha(k))\mathrm{u}(k) + \mathbf{B}_{\infty\omega}(\alpha(k))\omega_\infty(k) + \mathbf{B}_{2\omega}(\alpha(k))\omega_2(k), \\ \mathrm{z}_\infty(k) = \mathbf{C}_{\infty z}(\alpha(k))\mathrm{x}(k) + \mathbf{D}_{\infty u}(\alpha(k))\mathrm{u}(k) + \mathbf{D}_{\infty\omega}(\alpha(k))\omega_\infty(k), \\ \mathrm{z}_2(k) = \mathbf{C}_{2z}(\alpha(k))\mathrm{x}(k) + \mathbf{D}_{2u}(\alpha(k))\mathrm{u}(k), \\ \mathrm{y}(k) = \mathbf{C}_y\,\mathrm{x}(k) + \widetilde{\nu}(k), \end{cases} \tag{5.18}$$

where the parameter-dependent matrices have polytopic parametrization dependent on polytopic parameter $\alpha(k) \in \Lambda_{2n_\theta}$. Note that, when the open-loop system (5.18) is combined with a multi-objective gain-scheduled noisy static output-feedback (GSNOF) controller, the closed-loop system has a non-zero direct feed-forward term $\mathbf{D}_{2cl}(\alpha(k))$. To have finite $\mathcal{H}_2$ performance for $\mathrm{z}_2(\cdot)$, the direct feed-forward term needs to be zero; see [Skogestad and Postlethwaite, 2007] for more details. To satisfy this requirement, in this work a sensor filter is used for the measured output $\mathrm{y}(\cdot)$ and the control law (5.20) uses the filtered output $\widetilde{\mathrm{y}}(\cdot)$. Note that this technique is not conservative since practically each sensor signal needs to be filtered. Specifically, to remove redundancy, the details of the procedure steps will be explained through the simulation example. Together with, define the augmented states $\widetilde{\mathrm{x}}(\cdot) = \begin{bmatrix}\mathrm{x}^T(\cdot) & \widetilde{\mathrm{y}}^T(\cdot)\end{bmatrix}^T$, where $\mathrm{x}^T(\cdot) = [\mathrm{x}_1(\cdot)\cdots\mathrm{x}_n(\cdot)]^T \in \mathbb{R}^n$ is the states vector; $\widetilde{\mathrm{y}}^T(\cdot) = \begin{bmatrix}\widetilde{\mathrm{y}}_1(\cdot)\cdots\widetilde{\mathrm{y}}_q(\cdot)\end{bmatrix}^T \in \mathbb{R}^q$ is the filtered measured output vector. Consequently, system (5.18) is re-written as

$$\begin{cases} \widetilde{\mathrm{x}}(k+1) = \widetilde{\mathbf{A}}(\alpha(k))\, \widetilde{\mathrm{x}}(k) + \widetilde{\mathbf{B}}_u(\alpha(k))\, \mathrm{u}(k) + \widetilde{\mathbf{B}}_{2\omega}(\alpha(k))\, \widetilde{\omega}_{2p}(k) + \widetilde{\mathbf{B}}_{\infty\omega}(\alpha(k))\, \widetilde{\omega}_{\infty p}(k), \\ \widetilde{\mathrm{z}}_\infty(k) = \widetilde{\mathbf{C}}_{\infty z}(\alpha(k))\mathrm{x}(k) + \widetilde{\mathbf{D}}_{\infty u}(\alpha(k))\mathrm{u}(k) + \widetilde{\mathbf{D}}_{\infty\omega}(\alpha(k))\widetilde{\omega}_{\infty p}(k), \\ \widetilde{\mathrm{z}}_2(k) = \widetilde{\mathbf{C}}_{2z}(\alpha(k))\, \widetilde{\mathrm{x}}(k) + \widetilde{\mathbf{D}}_{2u}(\alpha(k))\, \mathrm{u}(k), \\ \widetilde{\mathrm{y}}(k) = \widetilde{\mathbf{C}}_y\, \widetilde{\mathrm{x}}(k), \end{cases} \tag{5.19}$$

where the parameter-dependent matrices have polytopic parametrization dependent on polytopic parameters vector $\alpha(k) \in \Lambda_{2n_\theta}$; $\widetilde{\mathrm{x}}(\cdot) = \left[\mathrm{x}^T(\cdot)\ \widetilde{\mathrm{y}}^T(\cdot)\right]^T$; $\widetilde{\omega}_{2p}(k) = \left[\omega_2^T(k)\ \widetilde{\nu}^T(k)\right]^T$; $\widetilde{\omega}_{\infty p}(k) = \left[\omega_\infty^T(k)\ \widetilde{\nu}^T(k)\right]^T$; $\widetilde{\mathbf{C}}_{2z}(\cdot) = \left[\mathbf{C}_{2z}(\cdot)\ \mathbf{0}\right]$; $\widetilde{\mathbf{C}}_{\infty z}(\cdot) = \left[\mathbf{C}_{\infty z}(\cdot)\ \mathbf{0}\right]$; $\widetilde{\mathbf{D}}_{\infty\omega}(\cdot) = \left[\mathbf{D}_{\infty\omega}(\cdot)\ \mathbf{0}\right]$; $\widetilde{\mathbf{D}}_{2u}(\cdot) = \mathbf{D}_{2u}(\cdot)$; $\widetilde{\mathbf{D}}_{\infty u}(\cdot) = \mathbf{D}_{\infty u}(\cdot)$; $\widetilde{\mathbf{C}}_y = \left[\mathbf{0_3}\ \ \mathbf{C}_y\right]$; and $\widetilde{\mathbf{A}}(\alpha(k))$, $\widetilde{\mathbf{B}}_u(\alpha(k))$, $\widetilde{\mathbf{B}}_{2\omega}(\alpha(k))$, $\widetilde{\mathbf{B}}_{\infty\omega}(\alpha(k))$ matrices structure will be explained in details in section 6.4 (to eliminate the redundancy) and other matrices are defined in Eqn. (5.18).

The multi-objective gain-scheduled noisy static output-feedback controller (GSNOF) is given by

$$\mathrm{u}(k) = \mathbf{K}(\alpha(k))\, \widetilde{\mathrm{y}}(k), \tag{5.20}$$

which can be further partitioned as follows

$$\mathrm{u}(k) = \left[\mathrm{u}_1(k), \mathrm{u}_2(k), \cdots, \mathrm{u}_m(k)\right]^T, \tag{5.21}$$

where $\mathbf{K}(\alpha(k))$ is the gain-scheduled control gain. When the open-loop system (5.19) is combined with a multi-objective gain-scheduled noisy static output-feedback (GSNOF) controller (5.20), the resulting closed-loop system can be written as follows

$$
\mathcal{CL}:\begin{cases}
\widetilde{\mathrm{x}}(k+1) = \mathbf{A}_{cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k) + \mathbf{B}_{2cl}(\alpha(k))\,\widetilde{\omega}_{2p}(k) + \mathbf{B}_{\infty cl}(\alpha(k))\,\widetilde{\omega}_{\infty p}(k),\\[2ex]
\widetilde{\mathrm{z}}_{\infty}(k) = \mathbf{C}_{\infty cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k) + \mathbf{D}_{\infty cl}(\alpha(k))\,\widetilde{\omega}_{\infty p}(k),\\[2ex]
\widetilde{\mathrm{z}}_{2}(k) = \mathbf{C}_{2cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k),
\end{cases} \tag{5.22}
$$

where,

$$
\begin{cases}
\mathbf{A}_{cl}(\alpha(k)) = \widetilde{\mathbf{A}}(\alpha(k)) + \widetilde{\mathbf{B}}_{u}(\alpha(k))\,\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_{y},\\[2ex]
\mathbf{B}_{2cl}(\alpha(k)) = \widetilde{\mathbf{B}}_{2\omega}(\alpha(k)),\\[2ex]
\mathbf{B}_{\infty cl}(\alpha(k)) = \widetilde{\mathbf{B}}_{\infty\omega}(\alpha(k)),\\[2ex]
\mathbf{C}_{\infty cl}(\alpha(k)) = \widetilde{\mathbf{C}}_{\infty z}(\alpha(k)) + \widetilde{\mathbf{D}}_{\infty u}(\alpha(k))\,\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_{y},\\[2ex]
\mathbf{C}_{2cl}(\alpha(k)) = \widetilde{\mathbf{C}}_{2z}(\alpha(k)) + \widetilde{\mathbf{D}}_{2u}(\alpha(k))\,\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_{y},\\[2ex]
\mathbf{D}_{\infty cl}(\alpha(k)) = \widetilde{\mathbf{D}}_{\infty\omega}(\alpha(k)).
\end{cases} \tag{5.23}
$$

Next, the desired $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performances are defined and control input covariance constraints are presented.

Definition 16 *Suppose that system* (5.22) *is exponentially stable, and let* $\mathcal{S}_{\widetilde{z}_\infty,\widetilde{\omega}_{\infty p}}$ *be its transfer function matrix from* $\widetilde{\omega}_{\infty p}$ *to* $\widetilde{z}_\infty$*. The* $\mathcal{H}_\infty$ *performance (for given* $\alpha(k)$*) for the LPV system* (5.22) *is given by*

$$
||\mathcal{S}_{\widetilde{z}_\infty,\widetilde{\omega}_{\infty p}}(\alpha(k))||_\infty = \sup_{||\widetilde{\omega}_{\infty p}(k)||_2\neq 0} \frac{||\widetilde{z}_\infty(k)||_2}{|\widetilde{\omega}_{\infty p}(k)||_2}, \tag{5.24}
$$

where, $\widetilde{\omega}_{\infty p}(k) \in \ell_2^{r_{\infty\omega}+r_{2\omega}+r_{\widetilde{v}}}$*; and* $\widetilde{z}_\infty(k) \in \ell_2^{p}$*. Then,* $\mathcal{H}_\infty$ *performance (for given* $\alpha(k)$*) is upper bounded by*

$$
\sup_{\alpha(k)\in\Lambda_{2n_\theta}} ||\mathcal{S}_{\widetilde{z}_\infty,\widetilde{\omega}_{\infty p}}(\alpha(k))||_\infty \;\leq\; \inf_{\mathbf{P}_\infty(\alpha(k)),\mathbf{G}(\alpha(k)),\eta} \eta \tag{5.25}
$$

where η *is a positive scalar number; and* $\mathbf{P}_\infty(\alpha(k)), \mathbf{G}(\alpha(k))$ *are designed matrices will be shown* *later in Lemma 21. Note that, the upper bound of the $\mathcal{H}_\infty$ performance* (5.25) *is sometimes called the upper bound of the worst cast norm from $\widetilde{\omega}_{\infty p}(k)$ to $\widetilde{z}_\infty(k)$.*

Definition 17 *Considering system* (5.19) *and suppose that system* (5.22) *is exponentially stable. The definition of the $\mathcal{H}_2$ performance for the LPV system* (5.22) *is given by*

$$||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 = \lim_{\mathcal{T}\to\infty} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} \mathbb{E}\left\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}} \widetilde{z}_2^T(k)\widetilde{z}_2(k)\right\}, \tag{5.26}$$

where Λ_{2n_θ} denotes the polytopic space, where the scheduling parameters belong to; $\mathbb{E}$ denotes the expectation operator; and the positive integer $\mathcal{T}$ denotes the time horizon. See Appendix B for more details.

Definition 18 *Considering system* (5.19)*, the constraints on the control input $u(k) = [u_1(k), \cdots, u_m(k)]^T$ in Eqn.* (5.20) *are bounded by*

$$Cov(u_r(k)) \leq \bar{\mathbf{U}}_r, \quad r = 1, \cdots, m, \tag{5.27}$$

where $\bar{\mathbf{U}}_r > 0$ is a given upper bound on the control covariance $Cov(u_r(k))$ for the r^{th} control input (i.e., $u_r(k)$). See Appendix E for more details.

Definition 19 *Considering system* (5.22)*, the trace of the output covariance (i.e., control input covariance constraints (ICC) cost function) is bounded by*

$$ICC\ cost = trace\Big(\mathbf{C}_{2cl}(\alpha(k))\ \mathbf{P}_{\mathbf{G}}(\alpha(k))\ \mathbf{C}_{2cl}^T(\alpha(k)\Big) < trace\Big(\mathbf{W}(\alpha(k))\Big), \tag{5.28}$$

where $\mathbf{W}(\alpha(k))$ is an upper bound of the ICC cost to be defined later in Eqns. (5.30) *and* (5.31)*;*

and $\mathbf{P_G}(\alpha(k))$ *is the solution to the time-varying Lyapunov equation*

$$\mathbf{P_G}(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P_G}(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\,\mathbf{B}_{2cl}^T(\alpha(k)), \qquad \mathbf{P_G}(0) = 0. \tag{5.29}$$

See Appendix .6 for more details. Note that, covariance matrices are $\mathbb{E}\left[\widetilde{v}(k)\widetilde{v}^T(j)\right] = \mathbf{I}$ and $\mathbb{E}\left[\omega_2(k)\omega_2^T(j)\right] = \mathbf{W}_{k_2}$, where the later cold be assumed with identity without conservativeness. Therefore, the last term of Eqn. (5.29) shows as $\mathbf{B}_{2cl}(\alpha(k))\,\mathbf{B}_{2cl}^T(\alpha(k))$. Next, the multi-objective control subject to constraints on the $\mathcal{H}_\infty$ performance and the control input covariance matrix is defined.

Problem 20 *Synthesize a multi-objective gain-scheduled noisy static output-feedback controller* (5.20) *for LPV system* (5.19) *that minimizes the output covariance performance upper bound* (5.28) *subject to the multiple constraints on the control inputs covariances* (5.27) *and the* $\mathcal{H}_\infty$ *performance* (5.25) *utilizing the estimated sensor noise covariance as a part of the gain-scheduling parameters.*

In other words, the goal of this work is to design a multi-objective noisy static output-feedback gain-scheduling controller that exponentially stabilizes the closed-loop system (5.22) and minimizes the upper bound of the performance output covariance (i.e., the trace of the performance output upper bound (5.28)) subject to constraints on the control inputs covariances ($\mathcal{H}_2$ constraints on control inputs) and the $\mathcal{H}_\infty$ performance. For more details about Input Covariance Constraints (ICC) control problem; see [Zhu et al., 1995], [White et al., 2013b] and therein references. Based on the bounded real lemma, an upper bound for the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performances can be computed using extend LMI characterization; see [De Caigny et al., 2012] for more details. The upper bounds of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performances of the closed-loop system (5.22) are given by the following lemma.

Lemma 21 *Consider the closed-loop system* (5.22)*, and the gain-scheduled controller* (5.20)*. The trace of the $\mathcal{H}_2$ output performance (output covariance), subject to constraints on the $\mathcal{H}_\infty$ performance and the control input covariance matrix, is upper bounded by $trace(\mathbf{W}(\alpha))$ if there exists a symmetric positive definite matrix $\mathbf{P}_2(\alpha(k)) = \mathbf{P}_2^T(\alpha(k))$ and matrices $\mathbf{G}(\alpha(k))$ and $\mathbf{W}(\alpha(k))$ such that the following matrix inequalities hold, for all $\alpha(k) \in \Lambda_{2n_\theta}$,*

$$\begin{bmatrix} \mathbf{P}_2(\alpha(k+1)) & * & * \\ \mathbf{G}^T(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) & \mathbf{G}(\alpha(k)) + \mathbf{G}^T(\alpha(k)) - \mathbf{P}_2(\alpha(k)) & * \\ \mathbf{B}_{cl}^T(\alpha(k)) & \mathbf{0} & \mathbf{I} \end{bmatrix} > 0, \tag{5.30}$$

$$\begin{bmatrix} \mathbf{W}(\alpha(k)) & * \\ \mathbf{G}^T(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k)) & \mathbf{G}(\alpha(k)) + \mathbf{G}^T(\alpha(k)) - \mathbf{P}_2(\alpha(k)) \end{bmatrix} > 0. \tag{5.31}$$

$$\begin{bmatrix} \bar{\mathbf{U}}_r & \mathbf{\Gamma}_r\mathbf{K}(\alpha(k))\widetilde{\mathbf{C}}_y \\ * & \mathbf{P}_2(\alpha(k)) \end{bmatrix} > 0, \quad \forall\, r = 1, \cdots, m, \tag{5.32}$$

where $\mathbf{\Gamma}_r$ is a given input channel section matrix for the control input r. In addition, the $\mathcal{H}_\infty$ performance output is bounded by η, if there is exist a symmetric positive definite matrix $\mathbf{P}_\infty(\alpha(k)) = \mathbf{P}_\infty^T(\alpha(k))$ and the same above $\mathbf{G}(\alpha)$ matrix, such that the following matrix inequality holds, for all $\alpha(k) \in \Lambda_{2n_\theta}$

$$\begin{bmatrix} \mathbf{P}_\infty(\alpha(k+1)) & * & * & * \\ \mathbf{G}^T(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) & \mathbf{G}(\alpha(k)) + \mathbf{G}^T(\alpha(k)) - \mathbf{P}_\infty(\alpha(k)) & * & * \\ \mathbf{B}_{cl}^T(\alpha(k)) & \mathbf{0} & \eta\mathbf{I} & * \\ \mathbf{0} & \mathbf{C}_{\infty cl}(\alpha(k))\mathbf{G}(\alpha(k)) & \mathbf{D}_{\infty cl}(\alpha(k)) & \eta\mathbf{I} \end{bmatrix} > 0. \tag{5.33}$$

Proof 22 *See Appendix C.*

There are some observations about the previous parametrized LMI (PLMI) set need to be mentioned. The previous PLMIs need to be solved for each $\alpha(k)$ in the polytopic space Λ_{2n_θ}, leading to an infinite set of LMIs (i.e., not tractable optimization problem). In addition, the parameter-dependent Lyapunov matrix depends on the current and the future value of the scheduling parameter vector (i.e., $\alpha(k)$ and $\alpha(k+1)$, respectively), the resulting controller not only depends on the current value of the scheduling parameter but also its future value $\alpha(k+1)$. Since in most applications, this value is not available in real-time, the gain-scheduled controller obtained by solving (if the solution exists) the previous PLMIs cannot be implemented in practice. Moreover, the substitution of the closed-loop matrices generates a solution from the closed-loop matrices and the Slack variable $\mathbf{G}(\cdot)$ which is very hard to handle. To elevate those issues, next Lemma states the LMIs conditions for solving Problem (20) such that the above issues are avoided. Indeed, to obtain a finite set of LMI conditions, the following parameter-dependent structure is imposed on the Lyapunov matrix $\mathbf{P}(\alpha(k))$ such that

$$\mathbf{P}(\alpha(k)) = \sum_{i=1}^{n_\alpha} \alpha_i(k)\mathbf{P}_i, \quad \alpha(k) \in \Lambda_{2n_\theta} \tag{5.34}$$

Considering the uncertainty set Ψ_a in (5.12), each $\alpha_i(k)$ and $\Delta_{\alpha_i}(k)$ are given by

$$\alpha_i(k) = \sum_{i=1}^{M} f_i^j \widetilde{\alpha}_j(k) \;\text{ and }\; \Delta_{\alpha_i}(k) = \sum_{i=1}^{M} h_i^j \widetilde{\alpha}_j(k), \tag{5.35}$$

such that

$$\bar{\mathbf{P}}(\widetilde{\alpha}(k)) = \sum_{j=1}^{M} \widetilde{\alpha}_j(k)\bar{\mathbf{P}}_j. \tag{5.36}$$

where $\bar{\mathbf{P}}_j = \sum_{i=1}^{n_\theta}\left(f_{i1}^j + f_{i2}^j\right)\mathbf{P}_i$ and $\widetilde{\alpha}_j(k)$ is the j^{th} component of the scheduling parameter vector $\widetilde{\alpha}(k) = \left[\widetilde{\alpha}_1(k), \cdots, \widetilde{\alpha}_M(k)\right]^T \in \Lambda_M$ in the uncertainty set Ψ_a, defined in Eqn. (5.12). Using the same structure for $\alpha(k)$ in (5.35), the system matrices in (5.23) are also converted to the new representation in terms $\widetilde{\alpha}(k) \in \Lambda_M$ as in (5.36). In addition, by considering Eqn. (5.35) with $\alpha(k+1) = \Delta_\alpha(k) + \alpha(k)$,

$$\tilde{\mathbf{P}}(\widetilde{\alpha}(k)) = \sum_{j=1}^{M} \widetilde{\alpha}_j(k)\tilde{\mathbf{P}}_j, \tag{5.37}$$

where $\tilde{\mathbf{P}}_j = \sum_{i=1}^{n_\theta} \left(\left(f_{i1}^j + h_{i1}^j\right) + \left(f_{i2}^j + h_{i2}^j\right)\right)\mathbf{P}_i$. Using parameterizations above, the next Lemma presents the finite-dimensional LMIs used to solve Problem 28.

Lemma 23 *Given input covariance constraints* $\bar{\mathbf{U}}_r$ $(r = 1, \cdots, m)$ *and a specified* $\mathcal{H}_\infty$ *performance upper bound* η*, considering system* (5.19) *and assuming that vectors* f^j *and* h^j *in Eqn.* (5.14) *are given. If there exist,* $\forall\, i = 1, \cdots, n_\theta$*, symmetric positive-definite matrices* $\mathbf{P}_{\infty,i} \in \mathbb{R}^{n\times n}$*,* $\mathbf{P}_{2,i} \in \mathbb{R}^{n\times n}$*, and matrices* $\mathbf{G}_i \in \mathbb{R}^{n\times n}$*,* $\mathbf{Z}_i \in \mathbb{R}^{m\times n}$*,* $\mathbf{W}_i \in \mathbb{R}^{p_2\times p_2}$ *such that,*
for $j = 1, \cdots, M$*,*

$$\begin{bmatrix} \tilde{\mathbf{P}}_{\infty,j} & * & * & * \\ \bar{\mathbf{G}}_i^T\bar{\mathbf{A}}_j^T + \bar{\mathbf{Z}}_i^T\bar{\mathbf{B}}_{uj}^T & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{\infty,j} & * & * \\ \bar{\mathbf{B}}_{\infty\omega,j}^T & \mathbf{0} & \eta\mathbf{I} & * \\ \mathbf{0} & \bar{\mathbf{C}}_{\infty z,j}\bar{\mathbf{G}}_i + \bar{\mathbf{D}}_{\infty u,j}\bar{\mathbf{Z}}_i & \bar{\mathbf{D}}_{\infty\omega,j} & \eta\mathbf{I} \end{bmatrix} > 0, \tag{5.38}$$

(5.39)

$$\begin{bmatrix} \tilde{\mathbf{P}}_{2,j} & * & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_j^T + \bar{\mathbf{Z}}_i^T \bar{\mathbf{B}}_{uj}^T & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{2,j} & * \\ \bar{\mathbf{B}}_{2\omega,j}^T & \mathbf{0} & \mathbf{I} \end{bmatrix} > 0, \tag{5.40}$$

$$\tag{5.41}$$

$$\begin{bmatrix} \sum_{i=1}^{n_\theta} \left(f_{i1}^j + f_{i2}^j \right) \mathbf{W}_i & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}_{2z,j}^T + \mathbf{Z}_i^T \bar{\mathbf{D}}_{2u,j}^T & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{2,j} \end{bmatrix} > 0, \tag{5.42}$$

$$\tag{5.43}$$

$$\begin{bmatrix} \bar{\mathbf{U}}_r & \mathbf{\Gamma}_r \bar{\mathbf{Z}}_i \bar{\mathbf{C}}_y \\ * & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{2,j} \end{bmatrix} > 0, \forall r = 1, \cdots, m, \tag{5.44}$$

and for $j = 1, \cdots, M-1$, *and* $\ell = j+1, \cdots, M$,

$$\begin{bmatrix} \mathbf{\Theta}_{1\infty} & * & * & * \\ \mathbf{\Theta}_{2\infty} & \mathbf{\Theta}_{3\infty} & * & * \\ \bar{\mathbf{B}}_{\infty\omega,j}^T + \bar{\mathbf{B}}_{\infty\omega,\ell}^T & \mathbf{0} & 2\eta\mathbf{I} & * \\ \mathbf{0} & \mathbf{\Theta}_{4\infty} & \bar{\mathbf{D}}_{\infty\omega,j} + \bar{\mathbf{D}}_{\infty\omega,\ell} & 2\eta\mathbf{I} \end{bmatrix} > 0, \tag{5.45}$$

$$\tag{5.46}$$

$$\begin{bmatrix} \boldsymbol{\Theta}_1 & * & * \\ \boldsymbol{\Theta}_2 & \boldsymbol{\Theta}_3 & * \\ \bar{\mathbf{B}}^T_{2\omega,j} + \bar{\mathbf{B}}^T_{2\omega,\ell} & \mathbf{0} & 2\mathbf{I} \end{bmatrix} > 0, \tag{5.47}$$

$$\tag{5.48}$$

$$\begin{bmatrix} \sum_{i=1}^{n_\theta} \left(f_{i1}^{j} + f_{i2}^{j}\right)\mathbf{W}_i + \sum_{i=1}^{n_\theta} \left(f_{i1}^{\ell} + f_{i2}^{\ell}\right)\mathbf{W}_i & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}^T_{2z,j} + \bar{\mathbf{Z}}_i^T \bar{\mathbf{D}}^T_{2u,j} + \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}^T_{2z,\ell} + \bar{\mathbf{Z}}_i^T \bar{\mathbf{D}}^T_{2u,\ell} & \boldsymbol{\Theta}_3 \end{bmatrix} > 0, \tag{5.49}$$

$$\tag{5.50}$$

$$\begin{bmatrix} \bar{\mathbf{U}}_r & \boldsymbol{\Gamma}_r \bar{\mathbf{Z}}_i \widetilde{\mathbf{C}}_y \\ * & \boldsymbol{\Theta}_3, \end{bmatrix} > 0, \quad for \;\; r = 1, \cdots, m, \tag{5.51}$$

Then, the parameter-dependent noisy static output-feedback gain, is given by

$$\begin{gathered} \mathbf{K}(\alpha(k)) = \hat{\mathbf{Z}}(\alpha(k))\,\hat{\mathbf{G}}(\alpha(k))^{-1}, \\ \hat{\mathbf{Z}}(\alpha(k)) = \sum_{i=1}^{n_\theta} \alpha_i(k)\,\mathbf{Z}_i, \quad \hat{\mathbf{G}}(\alpha(k)) = \sum_{i=1}^{n_\theta} \alpha_i(k)\,\mathbf{G}_i, \end{gathered} \tag{5.52}$$

that exponentially stabilizes system (5.22) *with guaranteed* $\mathcal{H}_2$ *performance (output covariance) upper bound* χ*, which is given by*

$$\chi^2 = \min_{\mathbf{P}_{\infty,i},\mathbf{P}_{2,i},\mathbf{W}_i,\mathbf{G}_i,\mathbf{Z}_i} \max_i \; trace\big(\mathbf{W}_i\big), \tag{5.53}$$

subject to both the $\mathcal{H}_\infty$ *performance upper bound* η *(*$\mathcal{H}_\infty$ *performance constrain) and the following* *control input constraints,*

$$Cov(u(k)) \leq \bar{\mathbf{U}}_r \quad \forall r = 1, \cdots, m, \tag{5.54}$$

where $\bar{\mathbf{U}}_r$ is a given upper bound on the control covariance matrix $Cov(u(k))$ for the r^{th} control input; $\mathbf{\Gamma}_r$ is a given input channel selection matrix for control input r;

$$\bar{\mathbf{P}}_{\infty,j} = \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{P}_{\infty,i}, \qquad \tilde{\mathbf{P}}_{\infty,j} = \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^j + h_{i1}^j \big) + \big(f_{i2}^j + h_{i2}^j \big) \Big) \mathbf{P}_{\infty,i},$$

$$\bar{\mathbf{P}}_{2,j} = \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{P}_{2,i}, \qquad \tilde{\mathbf{P}}_{2,j} = \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^j + h_{i1}^j \big) + \big(f_{i2}^j + h_{i2}^j \big) \Big) \mathbf{P}_{2,i},$$

$$\bar{\mathbf{G}}_i = \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{G}_i, \qquad \bar{\mathbf{Z}}_i = \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{Z}_i,$$

$$\mathbf{\Theta}_{1\infty} = \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^j + h_{i1}^j \big) + \big(f_{i2}^j + h_{i2}^j \big) \Big) \mathbf{P}_{\infty,i} + \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^\ell + h_{i1}^\ell \big) + \big(f_{i2}^\ell + h_{i2}^\ell \big) \Big) \mathbf{P}_{\infty,i},$$

$$\mathbf{\Theta}_{2\infty} = \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_\ell^T + \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_j^T + \bar{\mathbf{Z}}_i^T \bar{\mathbf{B}}_{u\ell}^T + \bar{\mathbf{Z}}_i^T \bar{\mathbf{B}}_{uj}^T,$$

$$\mathbf{\Theta}_{3\infty} = 2\bar{\mathbf{G}}_i + 2\bar{\mathbf{G}}_i^T - \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{P}_{\infty,i} - \sum_{i=1}^{n_\theta} \Big(f_{i1}^\ell + f_{i2}^\ell \Big) \mathbf{P}_{\infty,i},$$

$$\mathbf{\Theta}_{4\infty} = \bar{\mathbf{C}}_{\infty z,j} \bar{\mathbf{G}}_i + \bar{\mathbf{D}}_{\infty u,j} \bar{\mathbf{Z}}_i + \bar{\mathbf{C}}_{\infty z,\ell} \bar{\mathbf{G}}_i + \bar{\mathbf{D}}_{\infty u,\ell} \bar{\mathbf{Z}}_i,$$

$$\mathbf{\Theta}_1 = \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^j + h_{i1}^j \big) + \big(f_{i2}^j + h_{i2}^j \big) \Big) \mathbf{P}_{2,i} + \sum_{i=1}^{n_\theta} \Big(\big(f_{i1}^\ell + h_{i1}^\ell \big) + \big(f_{i2}^\ell + h_{i2}^\ell \big) \Big) \mathbf{P}_{2,i},$$

$$\mathbf{\Theta}_2 = \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_\ell^T + \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_j^T + \bar{\mathbf{Z}}_i^T \bar{\mathbf{B}}_{u\ell}^T + \bar{\mathbf{Z}}_i^T \bar{\mathbf{B}}_{uj}^T,$$

$$\mathbf{\Theta}_3 = 2\bar{\mathbf{G}}_i + 2\bar{\mathbf{G}}_i^T - \sum_{i=1}^{n_\theta} \Big(f_{i1}^j + f_{i2}^j \Big) \mathbf{P}_{2,i} - \sum_{i=1}^{n_\theta} \Big(f_{i1}^\ell + f_{i2}^\ell \Big) \mathbf{P}_{2,i},$$

Proof 24 *See Appendix. D.*

Next remark considers the case of robust controller.

Remark 25 *The robust noisy static output-feedback controller guaranteeing the upper bound of $\mathcal{H}_2$ performance (output covariance), subject to constraints on the $\mathcal{H}_\infty$ performance and the control input covariance matrix, can be found by forcing matrices $\hat{\mathbf{Z}}(\alpha(k))$ and $\hat{\mathbf{G}}(\alpha(k))$ to be parameter-independent.*

5.4 Simulation Results and Discussion

The numerical example presented in this section is to demonstrate the advantage of considering sensor performance degradation due to sensor aging during the control design, where the control input and $\mathcal{H}_\infty$ performance are subjected to given constraints. That is, the goal is to synthesize a multi-objective gain-scheduled noisy output-feedback (GSNOF) controller that minimizes the upper bound of the output $\mathcal{H}_2$ performance (performance output covariance) subject to constraints on the control input covariance matrix and the $\mathcal{H}_\infty$ performance upper bound, where the sensor performance degradation due to aging is also used by controller as a gain-scheduling parameter to guarantee target system stability and achieve the desired performances if possible. The synthesized controllers are controller gain-scheduled by a combination of the time-varying parameters and the estimated time-varying sensor measurement noise covariance, and in this work, it is called a fully gain-scheduled; controller gain-scheduled by the estimated time-varying sensor measurement noise covariance is called a partially gain-scheduled; and robust controller is not gain-scheduled. Lemma 23 is used with semi-definite programming (SDP) solver interfaced by the available parser to solve this problem.

The relative improvement (RI) is used as a measure to compare the performances of the gain-scheduled and robust controllers, and it is given by

$$RI = \frac{\mathcal{H}^R - \mathcal{H}^{GS}}{\mathcal{H}^R} \times 100\% \tag{5.55}$$

where $\mathcal{H}^{GS}$ and $\mathcal{H}^R$ are the $\mathcal{H}_2$ performance upper bounds for the designed gain-scheduled and robust controllers, respectively.

The numerical complexity [Boyd and Vandenberghe, 2004, Nesterov, 2013] of an optimization problem based on LMIs set is given by

$$complexity = log(\mathcal{L}\mathcal{V}^3) \tag{5.56}$$

where $\mathcal{L}$ is the number of raws in the LMIs set and $\mathcal{V}$ is the number of scalar variables in the LMI set. Note that, simulation results are created using MATLAB R2015 and semi-definite programming (SDP) solver SeDuMi [Sturm, 1999] interfaced by the parser YALMIP [Lofberg, 2004] with a computer equipped with an Intel Core i7 2.6 GHz Processor and 16 GB RAM. The considered example represents a state-of-art for the gain-scheduling control used in many published literature; see for example, [White et al., 2016] and therein references.

Example: consider a discrete-time multiple-input multiple-output (MIMO) LPV system (5.1) with the following state-space matrices

$$\left\{\begin{array}{l}\mathbf{A}(\theta(k)) = \begin{bmatrix} -2+\theta_1(k)+\theta_2(k) & 0 & -1 \\ 1 & -0.5 & 0 \\ 0 & 1 & -0.5 \end{bmatrix}, \mathbf{B}_u(\theta(k)) = \begin{bmatrix} 1+\theta_1(k)+\theta_2(k) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{B}_{\infty\omega} = \mathbf{B}_{2\omega} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{C}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{D}_{2u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C}_{2z} = \mathbf{I}_3, \end{array}\right. \tag{5.57}$$

where the time-varying parameters are assumed to have the following parameters variation bounds $\theta_1(k) \in [-0.1, 0.1]$, $\theta_2(k) \in \left[10^{-7}, 0.4\right]$. Note that, system (5.57) was originally used in [De Oliveira et al., 1999] and later used in [Apkarian et al., 2000] and [White et al., 2016], and it considered here with a slight revision to fit our design problem. The discrete-time LPV system (5.57) is converted to a discrete-time polytopic by solving matrices $\mathbf{A}(\cdot)$ and $\mathbf{B}(\cdot)$ at the vertices of the parameter space polytope of the parameters follow the procedure shown in [White et al., 2013c]. As mentioned earlier, to match with the strictly proper requirements for finite $\mathcal{H}_2$ norm (i.e., feasible $\mathcal{H}_2$ control) in this work, a filter is used for the measured output $\mathrm{y}(\cdot)$ and the control law (5.20) uses the filtered output $\widetilde{y}(\cdot)$, where the z-transform of the filtered measured output is given by

$$\widetilde{Y}(z) = G(z)\, Y(z), \tag{5.58}$$

where $Y(z)$ is the z-transform of the measured output vector $\mathrm{y}(\cdot)$; $\widetilde{Y}(z)$ is the z-transform of the filtered measured output vector $\widetilde{\mathrm{y}}(\cdot)$; and $G(z) = diag\,(G_i(z))$ is the filter transfer function matrix.

The filter transfer function of the i^{th} output channel is given by

$$G_i(z) = \frac{1 - e^{-aT}}{1 - e^{-aT} z^{-1}}, \qquad \forall i = 1, \cdots, q, \tag{5.59}$$

where a is the pole of the filter in the continuous-time domain and q denotes the number of measured output vector components. To explain the filtering procedure, for instance, consider the first component of the measured output vector (i.e., $\mathrm{y}_1(\cdot) = \mathbf{C}_{y1}\, \mathrm{x}(\cdot) + \widetilde{\nu}_1(\cdot)$), where $\mathbf{C}_{y1}$ is the first raw of the matrix $\mathbf{C}_y$, then

$$\widetilde{Y}_1(z) = \Big(\frac{1 - e^{-aT}}{1 - e^{-aT} z^{-1}}\Big) Y_1(z), \tag{5.60}$$

where $Y_1(z)$ is the z-transform of the $\mathrm{y}_1(\cdot)$. Substitute for $Y_1(z)$, manipulate and taking z-transform inverse, it leads

$$\widetilde{\mathrm{y}}_1(k+1) = (1 - e^{-aT})\, \mathbf{C}_{y1}\, \mathrm{x}(k+1) + e^{-aT}\, \widetilde{\mathrm{y}}_1(k) + (1 - e^{-aT})\, \widetilde{\nu}_1(k). \tag{5.61}$$

where $\widetilde{\mathrm{y}}_1(\cdot)$ is the first component of the vector $\widetilde{\mathrm{y}}(\cdot)$; $\widetilde{\nu}_1(k)$ is the first component of the measurement noise vector $\widetilde{\nu}(k) = \big[\widetilde{\nu}_1(k), \cdots, \widetilde{\nu}_q(k)\big]^T$; and other arguments are defined before. Generally, $\forall i = 1, \cdots, q$

$$\begin{aligned}
\widetilde{\mathrm{y}}_i(k+1) &= (1 - e^{-aT})\, \mathbf{C}_{yi}\, \mathrm{x}(k+1) + e^{-aT}\, \widetilde{\mathrm{y}}_i(k) + (1 - e^{-aT})\, \widetilde{\nu}(k), \\
&= (1 - e^{-aT})\, \mathbf{C}_{yi}\, \mathbf{A}(\cdot)\mathrm{x}(k) + (1 - e^{-aT})\, \mathbf{C}_{yi}\, \mathbf{B}_u(\cdot)u(k) \\
&+ (1 - e^{-aT})\, \mathbf{C}_{yi}\, \mathbf{B}_{\infty\omega}(\cdot)\omega_\infty(k) + (1 - e^{-aT})\, \mathbf{C}_{yi}\, \mathbf{B}_{2\omega}(\cdot)\omega_2(k) \\
&+ e^{-aT}\, \widetilde{\mathrm{y}}_i(k) + (1 - e^{-aT})\, \widetilde{\nu}(k+1),
\end{aligned} \tag{5.62}$$

where $\widetilde{y}_i(\cdot)$ is the i^{th} component of the vector $\widetilde{y}(\cdot)$; and $\mathbf{C}_{yi}$ is the i^{th} raw of the matrix $\mathbf{C}_y$. Together with, define the augmented states $\widetilde{x}(\cdot) = \left[x^T(\cdot) \;\; \widetilde{y}^T(\cdot)\right]^T$, as follow

$$\widetilde{x}(k+1) = \begin{bmatrix} x^T(k+1) \\ \\ \widetilde{y}^T(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}(\cdot) & 0 & 0 & \cdots & 0 \\ (1-e^{-aT})\,\mathbf{C}_{y1}\mathbf{A}(\cdot) & e^{-aT} & 0 & \cdots & 0 \\ (1-e^{-aT})\,\mathbf{C}_{y2}\mathbf{A}(\cdot) & 0 & e^{-aT} & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ (1-e^{-aT})\,\mathbf{C}_{yq}\mathbf{A}(\cdot) & 0 & \cdots & 0 & e^{-aT} \end{bmatrix}}_{\widetilde{\mathbf{A}}(\cdot)} \underbrace{\begin{bmatrix} x^T(k) \\ \\ \widetilde{y}^T(k) \end{bmatrix}}_{\widetilde{x}(k)}$$

$$+ \underbrace{\begin{bmatrix} \mathbf{B}_{\infty\omega}(\cdot) & \mathbf{B}_{2\omega}(\cdot) & \mathbf{0} \\ (1-e^{-aT})\mathbf{C}_{y1}\mathbf{B}_{\infty\omega}(\cdot) & (1-e^{-aT})\mathbf{C}_{y1}\mathbf{B}_{2\omega}(\cdot) & \\ \vdots & \vdots & (\mathbf{1}-\mathbf{e}^{-\mathbf{aT}}) \\ (1-e^{-aT})\mathbf{C}_{yq}\mathbf{B}_{\infty\omega}(\cdot) & (1-e^{-aT})\mathbf{C}_{yq}\mathbf{B}_{2\omega}(\cdot) & \end{bmatrix}}_{\widetilde{\mathbf{B}}_{\omega}(\cdot)} \underbrace{\begin{bmatrix} \omega_\infty^T(k) \\ \omega_2^T(k) \\ \widetilde{\nu}^T(k), \end{bmatrix}}_{\widetilde{\omega}_p(\cdot)} \tag{5.63}$$

$$+ \underbrace{\begin{bmatrix} \mathbf{B}_u(\cdot) \\ \\ (1-e^{-aT})\,\mathbf{C}_{y1}\,\mathbf{B}_u(\cdot) \\ \vdots \\ (1-e^{-aT})\,\mathbf{C}_{yq}\,\mathbf{B}_u(\cdot) \end{bmatrix}}_{\widetilde{\mathbf{B}}_u(\cdot)} u(k)$$

where $x^T(\cdot) = [x_1(\cdot)\cdots x_n(\cdot)]^T \in \mathbb{R}^n$ is the states vector; $\widetilde{y}^T(\cdot) = \left[\widetilde{y}_1(\cdot)\cdots\widetilde{y}_q(\cdot)\right]^T \in \mathbb{R}^q$ is the filtered measured output vector; $\mathbf{1}-\mathbf{e}^{-\mathbf{aT}} = \mathrm{diag}(1-e^{-aT}) \in \mathbb{R}^q$; $\mathbf{0} = \left[0\cdots 0\right]^T \in \mathbb{R}^q$; and $\mathbf{C}_{yi}$ $(i = 1,\cdots,q)$ is the i^{th} raw of the matrix $\mathbf{C}_y$.

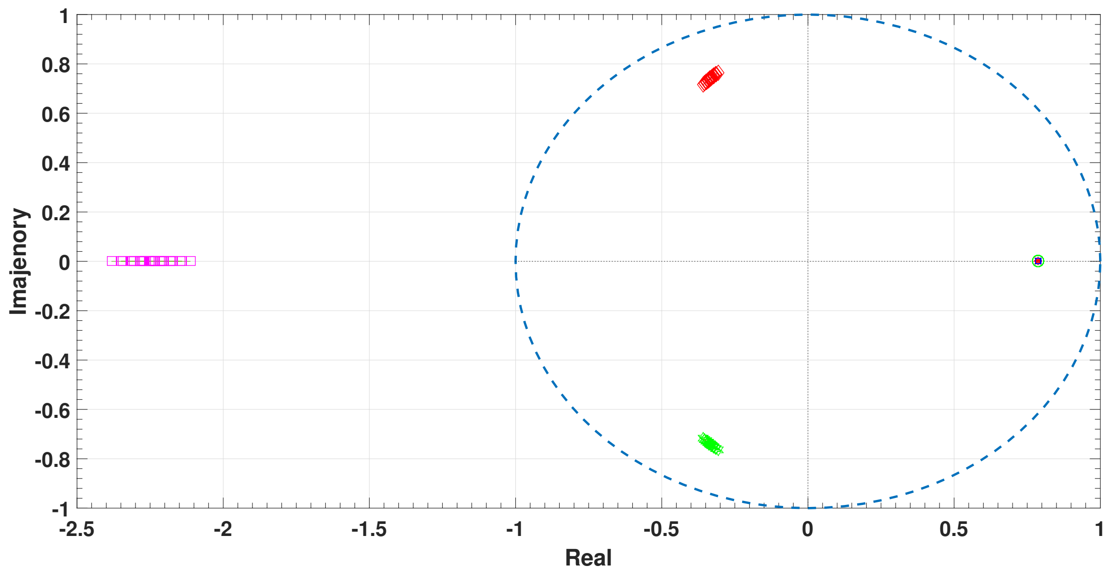


Figure 5.3: Poles location of open-loop system (5.57) considering design 1 for a wide range of the scheduling parameters (i.e., $\theta_1 \in \{-0.1, 0.1\}$ and $\theta_2 \in \{10^{-7}, 0.4\}$), where each pole locations are represented by different color and symbol.

For the current example (i.e., system (5.57)), $\mathbf{C}_y$ which has three raws, i.e., $q = 3$; $\mathrm{x}^T(\cdot) = [\mathrm{x}_1(\cdot)\,\mathrm{x}_2(\cdot)\,\mathrm{x}_3(\cdot)]^T \in \mathbb{R}^3$ is the states vector; $\widetilde{\mathrm{y}}^T(\cdot) = \left[\widetilde{\mathrm{y}}_1(\cdot)\,\widetilde{\mathrm{y}}_2(\cdot)\,\widetilde{\mathrm{y}}_3(\cdot)\right]^T \in \mathbb{R}^3$ is the filtered measured output vector; $\widetilde{\mathbf{D}}_{2u}(\cdot) = \mathbf{D}_{2u}(\cdot)$; and the filter parameters are set as $a = 0.8546$ and $T = 0.04336$.

In addition, the scheduling parameters ranges are set as $\theta_1 \in \{-0.1, 0.1\}$ and $\theta_2 \in \{10^{-7}, 0.4\}$; the $\mathcal{H}_\infty$ performance upper bound (η) is set to 20; and the input covariance constraints are set as

- Design 1 : $\mathrm{Cov}(\mathrm{u}_r(k)) \leq \bar{\mathbf{U}}_r, \quad \forall r = 1, 2, 3,$

- Design 2 : $\mathrm{Cov}(\mathrm{u}_r(k)) \leq \bar{\mathbf{U}}, \quad \forall r = 1, 2, 3,$

where $\bar{\mathbf{U}}_r$ is the r^{th} upper bound on the control covariance $\mathrm{Cov}(\mathrm{u}_r(k))$ for the r^{th} control input. That is, design 1 considers that each input channel has different upper bound on the control input covariance while design 2 considers that all control input channels have the same upper bound of the control input covariance.

Fig. 5.3 shows the locations of the open-loop system poles over a wide range of the scheduling parameters, where different color and symbol represents each pole locations. It is clear that the

Table 5.1: Location of closed-loop system considering design 1 and $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$ for a wide range of the scheduling parameters.

	$\theta_1 \in \{-0.1, 0.1\}$,	$\theta_1 \in \{-0.1, 0.1\}$,	$\theta_1 \in \{-0.1, 0.1\}$,	$\theta_1 \in \{-0.1, 0.1\}$,
	$\theta_2 \in \{10^{-7}, 10^{-4}\}$	$\theta_2 \in \{10^{-7}, 10^{-2}\}$	$\theta_2 \in \{10^{-7}, 0.2\}$	$\theta_2 \in \{10^{-7}, 0.4\}$
$pole_1$	$0.4316 + 0.0000i$	$0.3946 + 0.0000i$	$-0.4870 + 0.0000i$	$-0.6808 + 0.0000i$
$pole_2$	$0.0536 + 0.1330i$	$0.0499 + 0.1218i$	$-0.0872 + 0.1559i$	$-0.0988 + 0.2584i$
$pole_3$	$0.0536 - 0.1330i$	$0.0499 - 0.1218i$	$-0.0872 - 0.1559i$	$-0.0988 - 0.2584i$
$pole_4$	$-0.1880 + 0.0000i$	$-0.1700 + 0.0000i$	$0.1700 + 0.0000i$	$0.2934 + 0.0035i$
$pole_5$	$-0.1974 + 0.0121i$	$-0.1778 + 0.0100i$	$0.1617 + 0.0106i$	$0.2934 - 0.0035i$
$pole_6$	$-0.1974 - 0.0121i$	$-0.1778 - 0.0100i$	$0.1617 - 0.0106i$	$0.2917 + 0.0000i$

open-loop system has unstable poles over the range of the scheduling parameters.

For the stability analysis, the fully gain-scheduled controller is considered, where other controllers have similar behavior. Table. 5.1 shows the locations of the closed-loop system poles for different values of the scheduling parameters. It is clear that the closed-loop system poles are all located within a unit circle. Note that, the poles locations does not indicate system stability.

For the performance analysis, Fig. 5.4 shows the $\mathcal{H}_2$ performance output upper bound as a function of the sensor noise covariance for fully and partially gain-scheduled controllers and robust controller. The solid line with star and dotted line with diamond relate to the fully and partially, respectively, gain-scheduled controllers, and dotted line with circle relates to the robust controller. The x-axis is logarithmic in sensor noise covariance, and the y-axis is logarithmic for desired performance. Fig. 5.5 shows the relative improvement defined in Eqn. (5.55) for the designed fully (solid line with circle) and partially (dotted lines with cross) gain-scheduled controllers, where the x-axis is logarithmic in noise covariance, and y-axis is linear for relative improvement (%). Those figures have interesting interpretations: as the sensor gets aged (i.e., sensor noise covariance increase), the output performance to satisfy the given constraints gets worse. Besides, for a given sensor aging information (i.e., sensor measurement noise covariance),

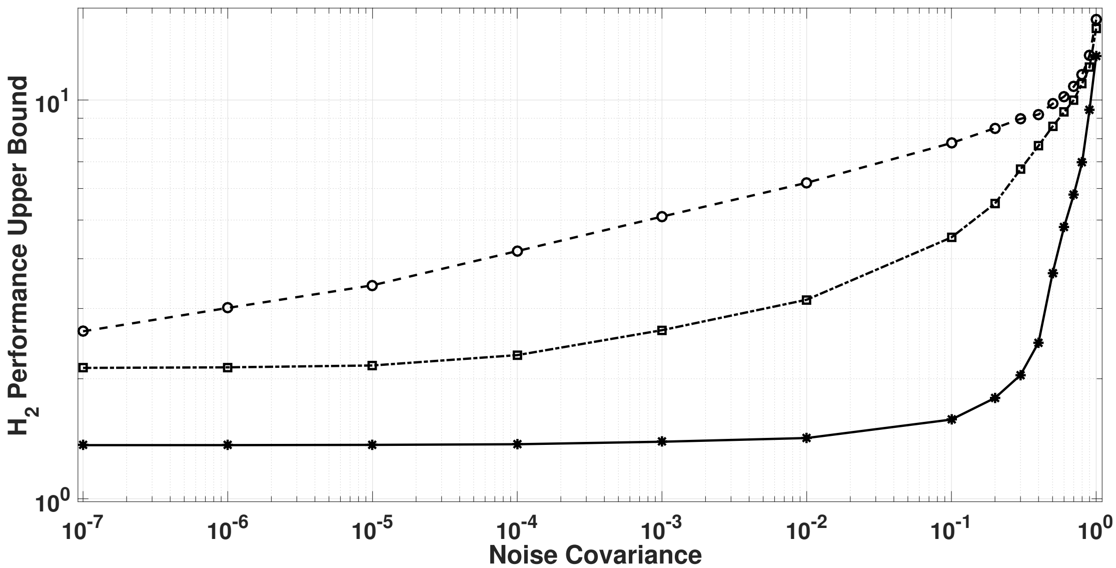


Figure 5.4: $\mathcal{H}_2$ performance upper bound as a function of the sensor noise covariance for the designed controllers using Lemma. 23, considering system (5.57) and design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$. The solid line with star marks and dotted line with diamond marks relate to the fully and partial, respectively, gain-scheduled controllers, and dotted line with circle marks relates to robust controller. The x-axis is logarithmic in sensor noise covariance and y-axis is logarithmic for desired performance.

the output performance upper bound can be found under the control energy limit (i.e., control input covariance constraint) effect. Moreover, the synthesized fully gain-scheduled controller compensates the effect of sensor performance with slow-degradation due to sensor aging (i.e., slowly-varying sensor noise covariance) and maintains the desired performance. On the other hand, the partially gain-scheduled controller attempts to compensate the effect of sensor aging for a while and then it starts to deviate from the desired performance. That is, closed-loop system $\mathcal{H}_2$ performance considering a partially gain-scheduled controller lies between fully gain-scheduled and robust controllers ones. Indeed, the designed fully gain-scheduled controller has the best performance over the partially gain-scheduled and robust controllers, and partially gain-scheduled has a worse performance than fully gain-scheduled but better performance than the robust one since the partially gain-scheduled control scheduled only by the sensor noise covariance and the robust controller is designed for a fixed gain. Finally, the closed-loop system performance with fully

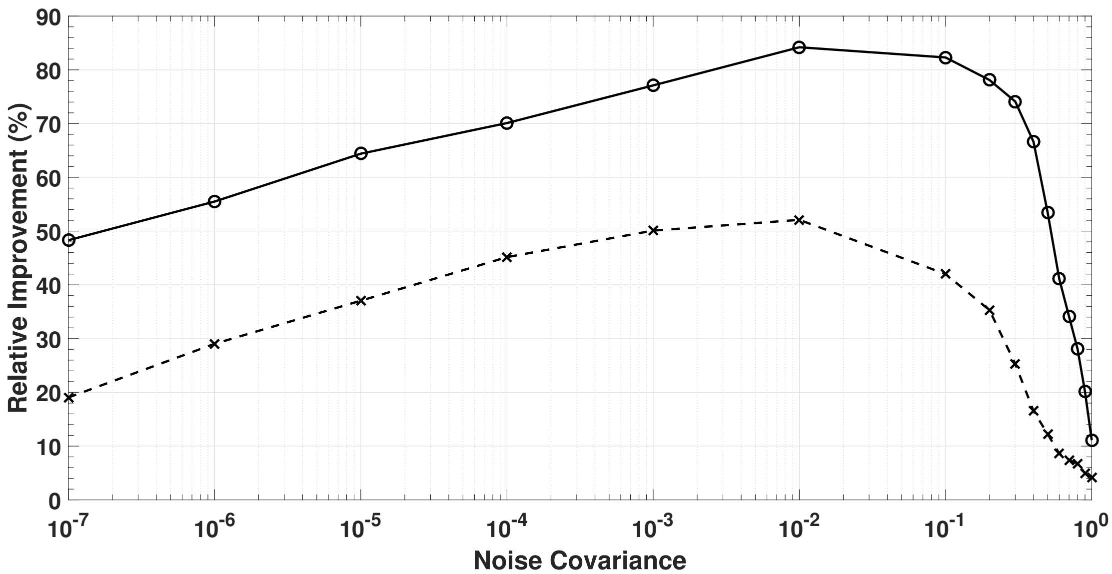


Figure 5.5: $\mathcal{H}_2$ performance relative improvement (Eqn. (5.55)) as a function of the sensor noise covariance for the designed fully (solid line with circle marks) and partially (dotted line with cross marks) gain-scheduled controllers over the robust controller using Lemma. 23, considering system (5.57) and design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$. The x-axis is logarithmic in noise covariance and y-axis is linear for relative improvement(%).

or partially gain-scheduled controls approaches the robust one with fairly large noise covariance. For the numerical complexity analysis, Table 5.2 presents the numerical complexity regarding the number of raws and number of scalar variables in the LMIs set using Lemma 23. Table 5.2 shows that fully gain-scheduled controller has the highest numerical complexity; partially gain-scheduled has mediated complexity; while the robust controller has the lowest one. Indeed, the fully gain-scheduled controller needs to solve much raws and scalar variables in the LMI set while robust controller solves for the minimum ones.

For the initial condition response analysis. Figs. 5.6a-5.7b show the time response for one state, to eliminate redundancy and get obviousness, considering fully gain-scheduled control; and design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$, where the sensor measurement noise covariance is set 10^{-4}, 10^{-2}, 0.2 and 0.4, respectively. Figs. 5.8a-5.9b show the time response, considering fully gain-scheduled control; and design 2 with $\bar{\mathbf{U}} = 0.01$, where the sensor measurement noise covariance

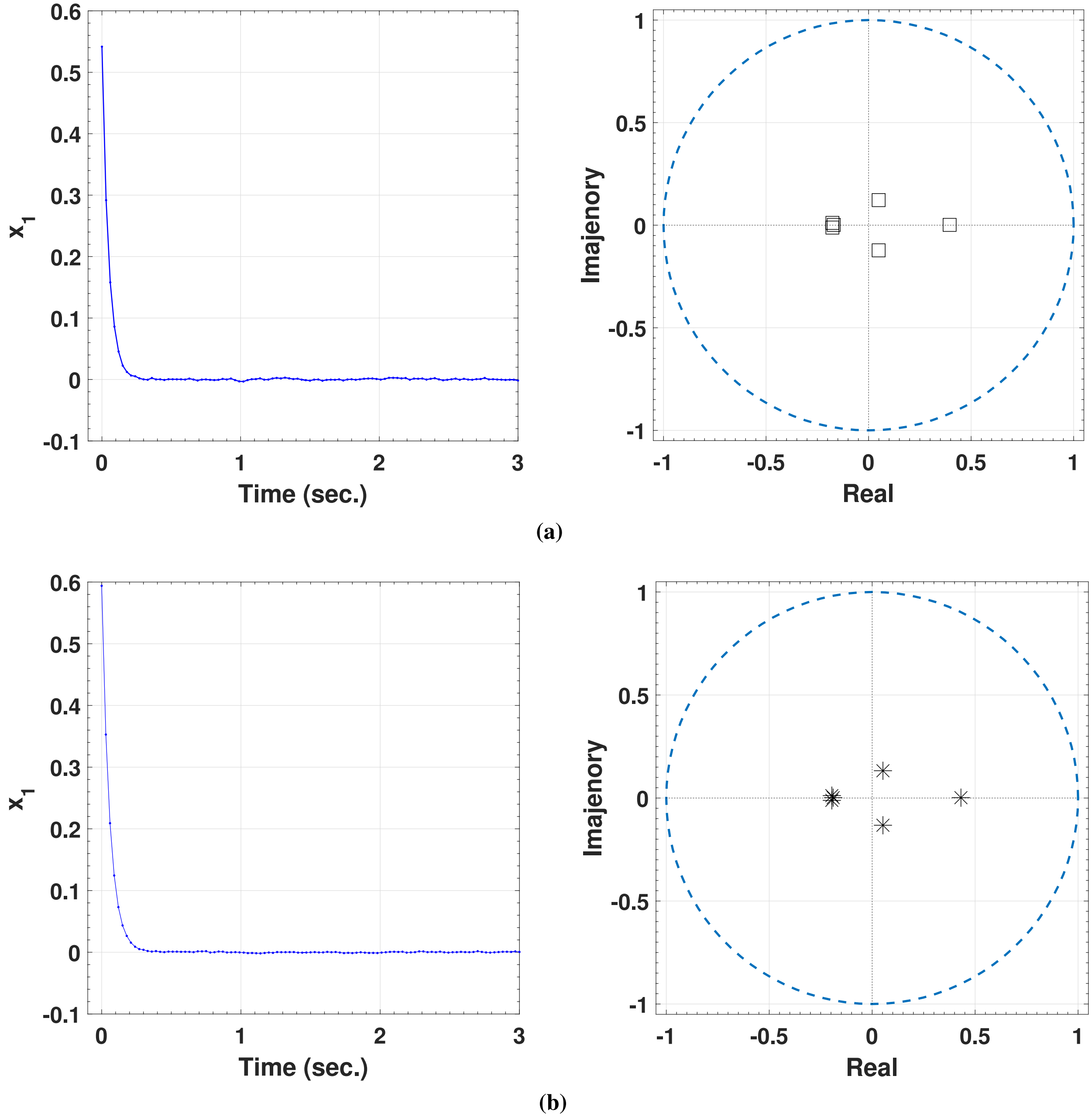
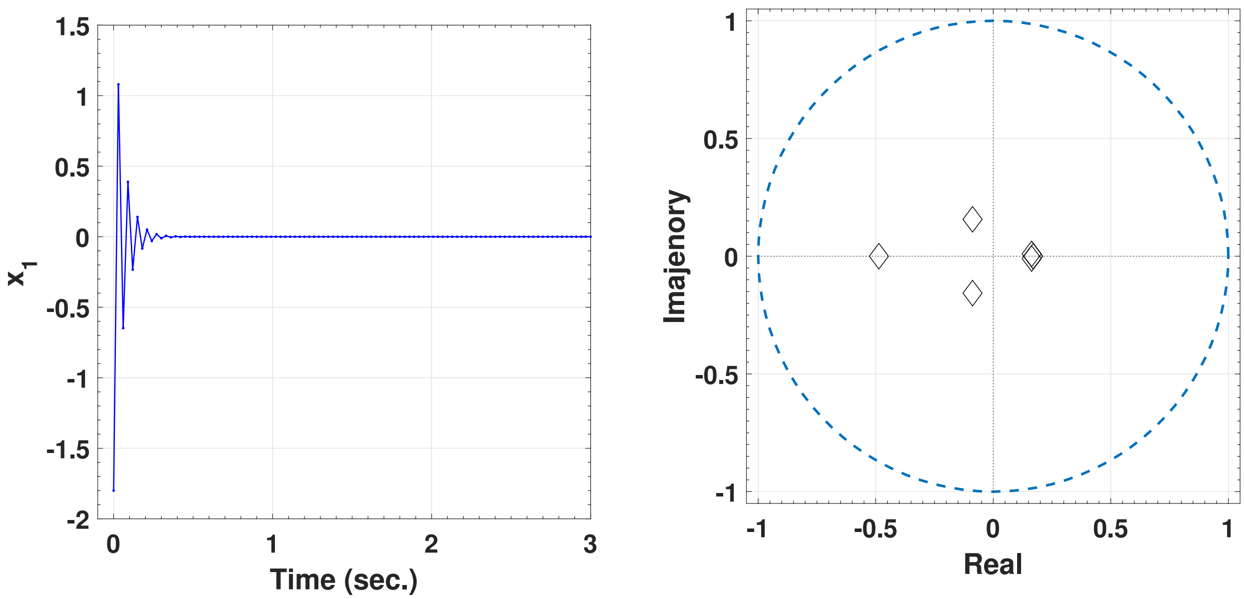
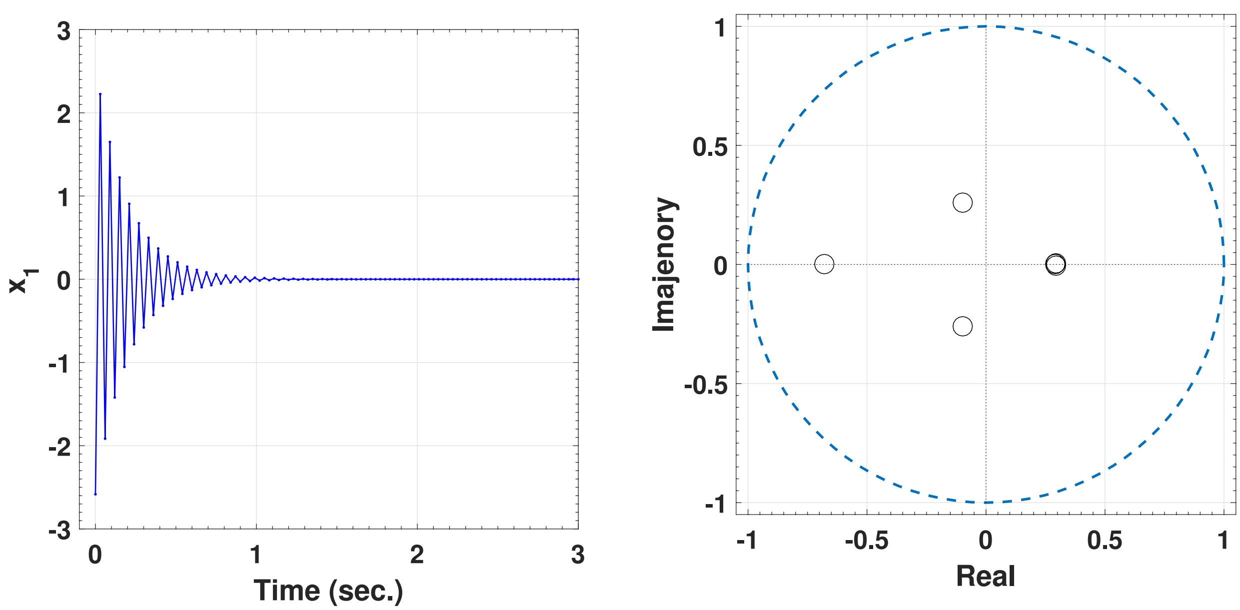


Figure 5.6: Closed-loop system initial condition response (left) and closed-loop system poles (right) for noise covariance (a) 10^{-4} (b) 10^{-2}, where design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$ is considered. x-axis is linear in time. y-axis is linear in system state magnitude.

is set 10^{-4}, 10^{-2}, 0.2 and 0.4, respectively. Those figures have interesting interpretations: as the sensor noise covariance increases, the time response gets slower. That is, the designed controller needs more time to compensate for the sensor noise effect. In addition, using a relaxed and the same input covariance constraint overall channels (i.e., design 2), the time response gets faster. That

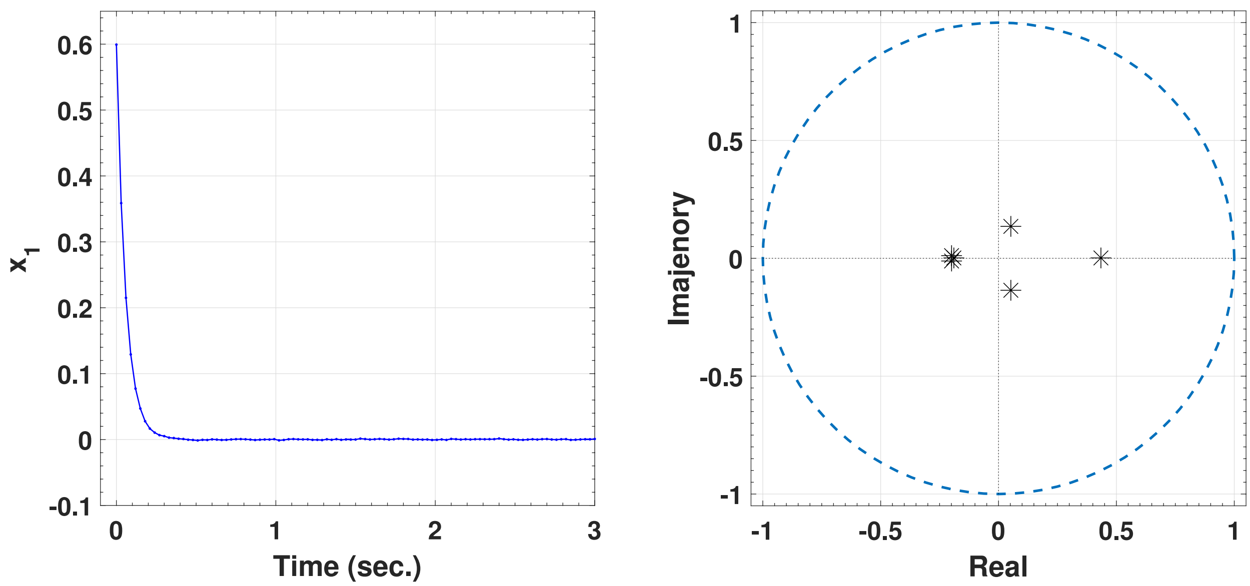


(a)

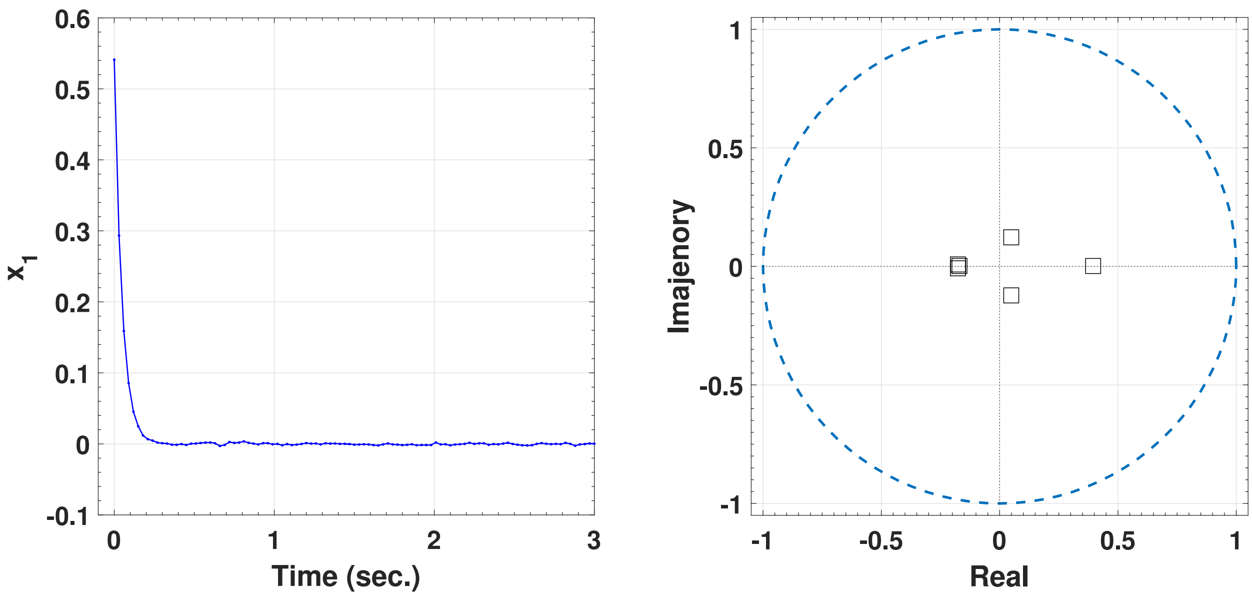


(b)

Figure 5.7: Closed-loop system initial condition response (left) and closed-loop system poles (right) for noise covariance (a) 0.2 (b) 0.4, where design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$ is considered. x-axis is linear in time. y-axis is linear in system state magnitude.



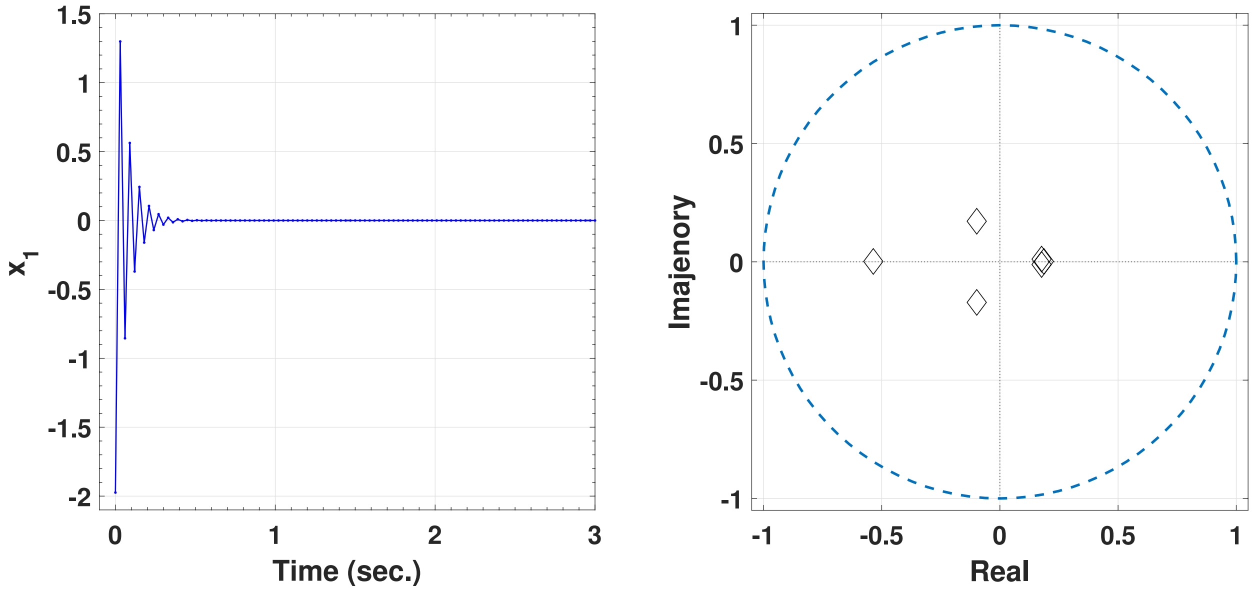
(a)



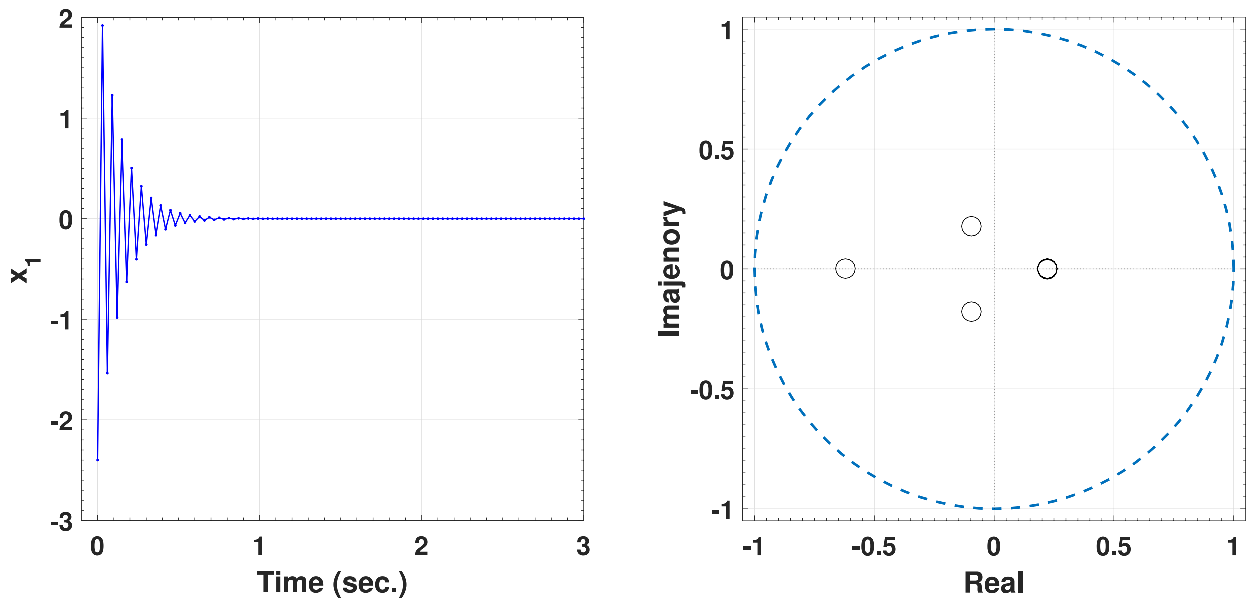
(b)

Figure 5.8: Closed-loop system initial condition response (left) and closed-loop system poles (right) for noise covariance (a) 10^{-4} (b) 10^{-2}, where design 2 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.01, 0.01, 0.01]^T$ is considered. x-axis is linear in time. y-axis is linear in system state magnitude.

is, the controller has a relatively high gain, then the system time response gets faster. Moreover, for tight input covariance constraints (at least over one channel), the designed controller leads to relatively slower response. That is, controller gain is tight and it leads to more time for response to settle down. Next, a summarization of the current work is presented.



(a)



(b)

Figure 5.9: Closed-loop system initial condition response (left) and closed-loop system poles (right) for noise covariance (a) 0.2 (b) 0.4, where design 2 with $\left[\bar{\mathrm{U}}_1, \bar{\mathrm{U}}_2, \bar{\mathrm{U}}_3\right]^T = [0.01, 0.01, 0.01]^T$ is considered. x-axis is linear in time. y-axis is linear in system state magnitude.

Table 5.2: Numerical complexity comparison between robust, partially and fully gain-scheduled control synthesized using Lemma. 23 and design 1 with $\left[\bar{\mathbf{U}}_1, \bar{\mathbf{U}}_2, \bar{\mathbf{U}}_3\right]^T = [0.005, 0.01, 0.015]^T$, where the sensor noise covariance is set to 0.2.

	$GS_{Lemma23}\ (full)$	$GS_{Lemma23}\ (partial)$	$Robust_{Lemma23}$
$\mathcal{L}$	$2,228,127$	$287,430$	$10,695$
$\mathcal{V}$	$1,442,559$	$745,370$	$33,323$
$\log(\mathcal{L}\mathcal{V}^3)$	57.16	53.13	40.51

5.5 Summary

Control system performance is heavily dependent on the sensor signals used for feedback control. A faulty sensor may lead to degraded system performance, an unstable system and/or even a fatal accident. In this paper, a multi-objective gain-scheduled control for linear parameter varying systems, subject to sensor performance degradation due to sensor aging, is presented, where the controller is subject to constraints on the control input variances and constraint on the $\mathcal{H}_\infty$ performance output. The time-varying sensor measurement noise covariance is used to characterize the sensor performance degradation due to sensor aging. Using the estimated sensor noise covariance, a multi-objective gain-scheduled noisy output-feedback controller is designed using the LPV technique. Additionally, there kinds of controllers are designed and they are controller scheduled by a combination of the time-varying parameters and the time-varying noise covariance (called fully gain-scheduled), controller scheduled by the sensor noise covariance (called partially gain-scheduled), and not scheduled (called robust controller). The synthesis of multi-objective gain-scheduled control is formulated as mixed ICC/$\mathcal{H}_\infty$ control problem that minimizes the performance output upper bound subject to $\mathcal{H}_2$ constraints on control input variances and $\mathcal{H}_\infty$ performance output. This problem can be solved efficiently using LMI optimization techniques. The proposed controller utilizes the sensor fault information (in terms of sensor noise covariance) to minimize its effect on the system and improves the closed-loop system performance. Also, the synthesized multi-objective gain-scheduled noisy output-feedback (GSNOF) controller compensates for the gradually-degraded sensor performance due to sensor aging and maintains the desired performance. The simulation results show that the synthesized fully gain-scheduled controller performs better than the partially gain-scheduled and robust one in terms of the performance with gradually-degraded sensor performance. In addition, the gain-scheduled controllers performance approach the robust one when the sensor noise covariance increases significantly.

Chapter 6

Guaranteed Performance Optimal Control for LPV Systems with Aging Sensors

6.1 Introduction

This chapter presents the design of a gain-scheduled controller that minimizes the control effort subject to performance constraint on output covariance matrix (OCC) in the presence of gradual sensor performance degradation due to sensor aging. The OCC control problem was first considered in [Zhu et al., 1993], then [Zhu et al., 1995] and [Zhu et al., 1997]. Then, it was presented in many literature for different applications such as linear time-invariant system [White et al., 2012], hydraulic engine cam phasing actuator [White et al., 2013a], and electric variable valve timing system [Ren and Zhu, 2011]. The OCC control problem is an optimal control problem, where the control effort is minimized subject to a constraint on output covariance matrix. The OCC problem has two interesting interpretations: stochastic and deterministic interpretations and the stochastic interpretation is used with this work. The stochastic interpretation is obtained by first assuming that the exogenous inputs are uncorrelated zero-mean white noises with known intensity (given or estimated). With this interpretation, the OCC problem is to minimize the control input covariance subject to the output covariance constraint(s). Indeed, in practical applications, it is often required for the designed controller to minimize the control effort subject to output performance constraint(s).

When closed-loop controllers are designed using the conventional methods, these constraints are usually not taken into account. On the other hand, with the OCC control design, the constraints on the regulated output are considered during the controller synthesis process. Therefore, the controller with the minimum control effort is obtained for a given set of constraints on the regulated output.

The main contributions of this chapter is a method using linear matrix inequalities (LMIs) to synthesize a gain-scheduled noisy output-feedback controller (GSNOF), utilizing the sensor aging information, such that the control effort is minimized subject to constraints on the regulated output covariance performance. Additionally, the sensor measurement noise covariance is used to model the sensor performance deterioration (fault). The gain-scheduled controller is scheduled by a combination of the time-varying parameters and the estimated sensor measurement noise covariance. The controller is validated in a simulation study for a discrete-time linear parameter varying (LPV) system with gradually performance degraded sensors; for more details see [Madhag and Zhu, 2019].

This chapter is organized as follows. Sub-section 6.2 provides an overview of the proposed problem, related mathematical preliminaries, and the discrete-time polytopic LPV system. Sub-section 6.3 synthesizing the LMI conditions for the proposed OCC control problem. Sub-section 6.4 provides the simulation validation results. Conclusions and future work are drawn in Sub-section 6.5.

6.2 Problem Definition

6.2.1 System Model

Consider a general polytopic discrete-time linear parameter varying (LPV) system described in the following state-space representation

$$
\begin{aligned}
\mathrm{x}(k+1) &= \mathbf{A}(\alpha(k))\mathrm{x}(k) + \mathbf{B}_u(\alpha(k))\mathrm{u}(k) + \mathbf{B}_{2\omega}(\alpha(k))\omega_2(k),\\
\mathrm{z}_2(k) &= \mathbf{C}_{2z}(\alpha(k))\mathrm{x}(k) + \mathbf{D}_{2u}(\alpha(k))\mathrm{u}(k),\\
\mathrm{y}(k) &= \mathbf{C}_y\,\mathrm{x}(k) + \widetilde{\nu}(k)
\end{aligned}
\tag{6.1}
$$

where $k \in \mathbb{Z}_+$ is the sample time index; $\mathrm{x}(k) \in \mathbb{R}^n$ is the state vector; $\mathrm{z}_2(k) \in \mathbb{R}^{p_2}$ is the system performance output vector; $\mathrm{y}(k) \in \mathbb{R}^q$ is the measured output vector; $\mathrm{u}(k) \in \mathbb{R}^m$ is the control input vector; $\mathbf{A}(\alpha(k)) \in \mathbb{R}^{n\times n}$; $\mathbf{B}_u(\alpha(k)) \in \mathbb{R}^{n\times m}$; $\mathbf{B}_{2\omega}(\alpha(k)) \in \mathbb{R}^{n\times r_{2\omega}}$; $\mathbf{C}_{2z}(\alpha(k)) \in \mathbb{R}^{p_2\times n}$; $\mathbf{C}_y \in \mathbb{R}^{q\times n}$; $\mathbf{D}_{2u}(\alpha(k)) \in \mathbb{R}^{p_2\times m}$; and the scheduling parameters vector $\alpha(k)$ is defined in Eqn. (6.3). The parameter-dependent matrices for system (6.1) are assumed to be polytopic with respect to the time-varying scheduling parameters vector. The initial state vector x(0) in system (6.1) is assumed to have a normal distribution with zero mean ($\mathbb{E}[\mathrm{x}(0)] = 0$, where $\mathbb{E}[\cdot]$ denotes the expectation operator) and covariance ($\mathbf{P}_0 = \mathbb{E}(\mathrm{x}(0)\mathrm{x}^T(0)) > 0$). System (6.1) is assumed to be observable and controllable. Note that, the observability and controllability assumptions are used mainly due to the fact that the observability and controllability for LPV systems are not well defined in the LPV control literatures. The terms $\omega_2(k) \in \mathbb{R}^{r_{2\omega}}$ and $\widetilde{\nu}(k) \in \mathbb{R}^{r_{\widetilde{\nu}}}$ in system (6.1) denote the process and measurement noises respectively. In addition, they are assumed to be uncorrelated Gaussian white ones with their means and covariances are defined by

$$\begin{cases} \mathbb{E}\left[\omega_2(k)\right] = 0,\ \mathbb{E}\left[\widetilde{\nu}(k)\right] = 0, \\ \mathbb{E}\left[\widetilde{\nu}(k)\widetilde{\nu}^T(j)\right] = \mathbf{I}\,\delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \\ \mathbb{E}\left[\omega_2(k)\omega_2^T(j)\right] = \mathbf{W}_{k_2}\,\delta(t_k - t_j) \quad \forall\, k, j \in \mathbb{R}, \\ \mathbb{E}\left[\omega_2(k)\widetilde{\nu}^T(j)\right] = \mathbb{E}\left[\omega_2(k)\mathrm{x}^T(j)\right] = \mathbb{E}\left[\widetilde{\nu}(k)\mathrm{x}^T(j)\right] = 0, \end{cases} \tag{6.2}$$

where δ is a Kronecker Delta function $\big($i.e., $\delta(0) = 1$ and $\delta(k) = 0; \forall\, k \neq 0\big)$. Note that $\widetilde{\nu}(k) = \mathbf{V}_k^{\frac{-1}{2}}\,\nu(k)$, $\mathbb{E}\left[\nu(k)\right] = 0$, $\mathbb{E}\left[\nu(k)\nu_j^T\right] = \mathbf{V}_k\,\delta(t_k - t_j), \forall\, k, j \in \mathbb{Z}_+$. The process noise covariance matrix, $\mathbf{W}_{k_2}$ is assumed to be positive semi-definite and known at each time instant and it can be assumed an indemnity matrix without conservativeness. Moreover, all sensor measurement noises are assumed to be independent (i.e., measurement noise for each channel is independent of the rest of them), that is, $\mathbf{V}_k = diag\big(\,\sigma_1(k), \cdots, \sigma_q(k)\,\big)$, where $\mathbf{V}_k$ is the sensor measurement noise covariance matrix and $\sigma_i(k)$ is the sensor measurement noise variance of the i^{th} channel. Define a scheduling parameters vector as

$$\alpha(k) = \left[\,\alpha_1(k), \cdots, \alpha_q(k),\ \alpha_{q+1}(k), \cdots, \alpha_{n_\alpha}(k)\,\right]^T, \tag{6.3}$$

where $\alpha_1(k), \cdots, \alpha_q(k)$ are due to sensors measurements noises covariance variation; $\alpha_{q+1}(k), \cdots, \alpha_{n_\alpha}(k)$ are due to system parameter variations and q is the number of measured output sensors. In addition, the parameters $\alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\alpha}(k)\right]^T$ are assumed to lie in a known polytopic space Λ_{n_α}; the vertices set of Λ_{n_α} is $ver(\Lambda_{n_\alpha}) = \{\eta_1, \cdots, \eta_{n_\alpha} |\ \eta_i \in \{\underline{\alpha}_i, \overline{\alpha}_i\}\}$. Note that, the polytopic space, Λ_{n_α}, could be defined as Λ_{2n_θ} if the polytopic system is considered as transformed version of the affine one following the change of variables in Eqn. (5.7). The scheduling parameters are assumed to be independent and time-varying parameters with the following known bounds

$$\underline{\alpha}_i \leq \alpha_i(k) \leq \overline{\alpha}_i, \qquad \forall\, i = 1, \cdots, n_\alpha \quad \forall\, k \in \mathbb{Z}_+, \tag{6.4}$$

where $\underline{\alpha}_i$ and $\overline{\alpha}_i$ are the lower and upper bounds of the i^{th} scheduling parameter, respectively. In addition, $\forall\ k \geq 0$, the rate of variation of the scheduling parameters is given by

$$\Delta_{\alpha_i}(k) = \alpha_i(k+1) - \alpha_i(k), \quad \forall\, i = 1, \cdots, n_\alpha, \tag{6.5}$$

and the bounds of the scheduling parameters variation rate are

$$\underline{\Delta}_{\alpha_i} \leq \Delta_{\alpha_i}(k) \leq \overline{\Delta}_{\alpha_i}, \quad \forall\, i = 1, \cdots, n_\alpha,\ \forall\, k \in \mathbb{Z}_+, \tag{6.6}$$

where $\overline{\Delta}_{\alpha_i}$ and $\underline{\Delta}_{\alpha_i}$ are the lower and upper bounds of the rate of variation of the i^{th} scheduling parameter, respectively. Without loss of generality, bounds in Eqn. (6.4) and Eqn. (6.6) are assumed to be symmetric. That is, variation rates bounds as $\left|\Delta_{\alpha_i}\right| \leq a_i,\ a_i \in \{0, 1\},\ \forall i = 1, \cdots, n_\alpha$, which represents the case where the scheduling parameters variate within a bounded rate of variation. Generally, each scheduling parameter $\alpha_i(k)\ \ (i = 1, \cdots, n_\alpha)$ belongs to Λ_{n_α}, it yields

$$\sum_{i=1}^{n_\alpha} \Delta_{\alpha_i} = 0, \tag{6.7}$$

where polytopic space parameters vector $\alpha(k) = \left[\alpha_1(k), \cdots, \alpha_{n_\alpha}(k)\right]^T \in \Lambda_{n_\alpha}$ and the parameters variation rates vector is $\Delta_\alpha(k) = \left[\Delta_{\alpha_1}(k), \cdots, \Delta_{\alpha_{n_\alpha}}(k)\right]^T \in \Lambda_{n_\alpha}$. In addition, in the discrete-time case, the rate of variation depends on the associated value of the parameter, which is the main difference from the continuous-time case, and assuming parameter rate independent could be very conservative, and for more details; see [Lacerda et al., 2016], [De Caigny et al., 2010] and [Oliveira and Peres, 2009]. The uncertainty domain, where the vector $\left[\alpha^T(k), \Delta_\alpha^T(k)\right]^T \in \Lambda_{n_\alpha}$ takes values, can be modeled by the compact set below [De Caigny et al., 2010]

$$\begin{aligned}\Psi_a = \Bigg\{ & \psi \in \mathbb{R}^{2n_\alpha} : \psi \in co\{\mu^1, \cdots, \mu^M\}, \quad \mu^j = \begin{bmatrix} f^j \\ h^j \end{bmatrix}, \\ & f^j \in \mathbb{R}^{n_\alpha}, \quad h^j \in \mathbb{R}^{n_\alpha}, \sum_{i=1}^{n_\alpha} f_i^j = 1, \quad f_i^j, \geq 0, \\ & \sum_{i=1}^{n_\alpha} h_i^j = 0, \quad \forall j = 1, \cdots, M, \quad \forall i = 1, \cdots, n_\alpha \Bigg\},\end{aligned} \tag{6.8}$$

as a convex combination of vectors μ^j for $j = 1, \cdots, M$, where vectors f_i^j and h_i^j are given a priori; and M is the numbers of vectors μ^j, which is given by [De Caigny et al., 2010], [White et al., 2013c]

$$M = N^2 + (N-1)^2 + (N-1), \tag{6.9}$$

where N is the number of the vertices of the polytopic system (i.e., number of the scheduling parameters). For more details about Eqn. (6.9) see [De Caigny et al., 2010], [White et al., 2013c] and therein references.

This definition of the set Ψ_a ensures that $\alpha(k)$ and Eqn. (6.7) hold for all $k \geq 0$. In the next section, the controller synthesizes using is presented.

6.3 Controller Synthesis LMIs

The goal of this work is to design a noisy static output-feedback gain-scheduling controller that exponentially stabilizes the closed-loop system and minimizes the control input covariance subject to constraint on the regulated output covariance. This section provides the synthesis LMI conditions for the proposed OCC control problem. To make the problem tractable, the upper bound of $\mathcal{H}_2$ control effort are minimized instead subject to constraint on the output covariance. Note that the closed-loop system has a non-zero direct feed-forward term $\mathbf{D}_{2cl}(\alpha(k))$. To have finite $\mathcal{H}_2$ performance for $\mathrm{z}_2(\cdot)$, it needs to be zero; see [Skogestad and Postlethwaite, 2007] for more details. To satisfy this requirement, in this work a dynamic sensor signal filter is used for the measured output $\mathrm{y}(\cdot)$ and the control law uses the filtered output $\widetilde{y}(\cdot)$. Note that this technique is not conservative since practically each sensor signal needs to be filtered in practical applications. Specifically, to remove redundancy, the details of the procedure steps will be explained through the simulation example. Together with, define the augmented states $\widetilde{\mathrm{x}}(\cdot) = \begin{bmatrix}\mathrm{x}^T(\cdot) & \widetilde{y}^T(\cdot)\end{bmatrix}^T$, where $\mathrm{x}^T(\cdot) = [\mathrm{x}_1(\cdot)\cdots\mathrm{x}_n(\cdot)]^T \in \mathbb{R}^n$ is the states vector; $\widetilde{y}^T(\cdot) = \begin{bmatrix}\widetilde{y}_1(\cdot)\cdots\widetilde{y}_q(\cdot)\end{bmatrix}^T \in \mathbb{R}^q$ is the filtered measured output vector. Consequently, system (6.1) is re-written as

$$\begin{cases} \widetilde{\mathrm{x}}(k+1) = \widetilde{\mathbf{A}}(\alpha(k))\,\widetilde{\mathrm{x}}(k) + \widetilde{\mathbf{B}}_u(\alpha(k))\,\mathrm{u}(k) + \widetilde{\mathbf{B}}_\omega(\alpha(k))\,\widetilde{\omega}_p(k), \\ \quad\widetilde{\mathrm{z}}_2(k) = \widetilde{\mathbf{C}}_{2z}(\alpha(k))\,\widetilde{\mathrm{x}}(k) + \widetilde{\mathbf{D}}_{2u}(\alpha(k))\,\mathrm{u}(k), \\ \quad\widetilde{\mathrm{y}}(k) = \widetilde{\mathbf{C}}_y\,\widetilde{\mathrm{x}}(k), \end{cases} \tag{6.10}$$

where the parameter-dependent matrices have polytopic parametrization dependent on polytopic parameter $\alpha(k) \in \Lambda_{n_\alpha}$; $\widetilde{\mathbf{C}}_{2z}(\cdot) = \begin{bmatrix}\mathbf{C}_{2z}(\cdot) & \mathbf{0}\end{bmatrix}$; $\widetilde{\mathbf{D}}_{2u}(\cdot) = \mathbf{D}_{2u}(\cdot)$; $\widetilde{\mathbf{C}}_y = \begin{bmatrix}\mathbf{0_3} & \mathbf{C}_y\end{bmatrix}$; $\widetilde{\mathrm{x}}(\cdot) = \begin{bmatrix}\mathrm{x}^T(\cdot) & \widetilde{y}^T(\cdot)\end{bmatrix}^T$; $\widetilde{\omega}_p(\cdot) = \begin{bmatrix}\omega_2^T(\cdot) & \widetilde{\nu}^T(\cdot)\end{bmatrix}^T$; other matrices defined in Eqn. (6.1) and $\widetilde{\mathbf{A}}(\alpha(k))$, $\widetilde{\mathbf{B}}_u(\alpha(k))$, $\widetilde{\mathbf{B}}_\omega(\alpha(k))$ matrices structure will explained in details at the simulation section. The gain-scheduled noisy static output-feedback controller (GSNOF), is given by

$$\mathrm{u}(k) = \mathbf{K}(\alpha(k))\,\widetilde{\mathrm{y}}(k), \tag{6.11}$$

where $\mathbf{K}(\alpha(k))$ is the gain-scheduled control gain. When the open-loop system (6.10) is combined with a gain-scheduled noisy static output-feedback (GSNOF) controller (6.11), the resulting closed-loop system can be written as follows

$$\mathcal{CL}: \begin{cases} \widetilde{\mathrm{x}}(k+1) = \mathbf{A}_{cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k) + \mathbf{B}_{cl}(\alpha(k))\,\widetilde{\omega}_p(k), \\ \widetilde{\mathrm{z}}_2(k) = \mathbf{C}_{2cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k), \end{cases} \tag{6.12}$$

where,

$$\begin{cases} \mathbf{A}_{cl}(\alpha(k)) = \widetilde{\mathbf{A}}(\alpha(k)) + \widetilde{\mathbf{B}}_u(\alpha(k))\,\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y, \\ \mathbf{B}_{cl}(\alpha(k)) = \widetilde{\mathbf{B}}_\omega(\alpha(k)), \\ \mathbf{C}_{2cl}(\alpha(k)) = \widetilde{\mathbf{C}}_{2z}(\alpha(k)) + \widetilde{\mathbf{D}}_{2u}(\alpha(k))\,\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y, \\ \widetilde{\omega}_p(k) = \begin{bmatrix} \omega_2^T(k) & \widetilde{\nu}^T(k) \end{bmatrix}^T. \end{cases} \tag{6.13}$$

Next, definitions of system control input covariance constraints and OCC cost function are presented.

Definition 26 *Considering system* (6.12)*, the constraints on the output covariance in Eqn.* (6.12) *(i.e., regulated output covariance constraint) are bounded by*

$$Cov(\widetilde{z}_2(k)) = \mathbf{C}_{2cl}(\alpha(k))\;\mathbf{P}_{\mathbf{G}}(\alpha(k))\;\mathbf{C}_{2cl}^T(\alpha(k) \leq \bar{Z}, \tag{6.14}$$

where $\bar{Z}$ is a given upper bound on the output covariance matrix $Cov(z(k))$ and $\mathbf{P}_{\mathbf{G}}(\alpha(k))$ is the solution to the time-varying Lyapunov equation

$$\mathbf{P}_{\mathbf{G}}(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_{\mathbf{G}}(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{cl}(\alpha(k))\;\mathbf{B}_{cl}^T(\alpha(k))\,. \tag{6.15}$$

Definition 27 *Considering system* (6.12)*, the input covariance constraint cost function (i.e., control* *energy covariance)* $\mathcal{U}_{OCC}$ *is given by*

$$\mathcal{U}_{OCC} = trace\Big(\mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y\,\mathbf{P}_{\mathbf{G}}(\alpha(k))\,\widetilde{\mathbf{C}}_y^T\,\mathbf{K}^T(\alpha(k))\Big), \tag{6.16}$$

Note that, the proof of Eqns.(6.14)-(6.16) can be obtained using operator theory by following the steps provided in [Zhu and Skelton, 1995].

Next, the gain-scheduled control subject to constraint on the output covariance (OCC) problem is defined.

Problem 28 *Synthesize a gain-scheduled noisy static output-feedback controller* (6.11) *for LPV system* (6.10) *that minimizes the control effort* (6.16) *subject to the constraint on the output covariance matrix* (6.14) *and utilize the estimated sensor noise covariance as part of the gain-scheduling parameters.*

For more details about output covariance constraints (OCC) control problem; see [Zhu et al., 1995], [Zhu and Skelton, 1995] and therein references. Problem 28 can be solved by performing a convex optimization over a set of linear matrix inequalities (LMIs). The LMIs list here are an extension of the work presented in [De Caigny et al., 2010]. To obtain a finite set of LMI conditions, the following parameter-dependent structure is imposed on the Lyapunov matrix $\mathbf{P}(\alpha(k))$ such that

$$\mathbf{P}(\alpha(k)) = \sum_{i=1}^{n_\alpha} \alpha_i(k)\mathbf{P}_i, \quad \alpha(k) \in \Lambda_{n_\alpha} \tag{6.17}$$

Considering the uncertainty set Ψ_a in (6.8), each $\alpha_i(k)$ and $\Delta_{\alpha_i}(k)$ are given by

$$\alpha_i(k) = \sum_{i=1}^{M} f_i^j \widetilde{\alpha}_j(k) \;\text{ and }\; \Delta_{\alpha_i}(k) = \sum_{i=1}^{M} h_i^j \widetilde{\alpha}_j(k), \tag{6.18}$$

such that

$$\bar{\mathbf{P}}(\widetilde{\alpha}(k)) = \sum_{j=1}^{M} \widetilde{\alpha}_j(k)\bar{\mathbf{P}}_j. \tag{6.19}$$

where $\bar{\mathbf{P}}_j = \sum_{i=1}^{n_\alpha} f_i^j \mathbf{P}_i$ and $\widetilde{\alpha}_j(k)$ is the j^{th} component of the scheduling parameter vector $\widetilde{\alpha}(k) = \left[\widetilde{\alpha}_1(k), \cdots, \widetilde{\alpha}_M(k)\right]^T \in \Lambda_M$ in the uncertainty set Ψ_a, that is defined in Eqn. (6.8). Using the same structure for $\alpha(k)$ in (6.18), the system matrices in (6.13) are also converted to the new representation in terms $\widetilde{\alpha}(k) \in \Lambda_M$ as in (6.19). In addition, considering Eqn. (6.18) with $\alpha(k+1) = \Delta_\alpha(k) + \alpha(k)$ leads to

$$\widetilde{\mathbf{P}}(\widetilde{\alpha}(k)) = \sum_{j=1}^{M} \widetilde{\alpha}_j(k)\widetilde{\mathbf{P}}_j, \tag{6.20}$$

where $\widetilde{\mathbf{P}}_j = \sum_{i=1}^{n_\alpha} \left(f_i^j + h_i^j\right)\mathbf{P}_i$. Using these parameterizations, next Lemma presents the finite-dimensional LMIs that can be used to solve Problem 28.

Lemma 29 *Given output covariance constraint upper bound $\bar{Z}$, considering system* (6.10) *and assuming that vectors f^j and h^j are given. If there exist, $\forall\, i = 1, \cdots, n_\alpha$, symmetric positive-definite matrices $\mathbf{P}_{2,i} \in \mathbb{R}^{n\times n}$, $\mathbf{W}_i \in \mathbb{R}^{p_2\times p_2}$ and matrices $\mathbf{G}_i \in \mathbb{R}^{n\times n}$, $\mathbf{X}_i \in \mathbb{R}^{p_2\times n}$, such that, for $j = 1, \cdots, M$,*

$$\begin{bmatrix} \widetilde{\mathbf{P}}_{2,j} & * & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_j^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{B}}_{uj}^T & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{2,j} & * \\ \bar{\mathbf{B}}_{2\omega,j}^T & \mathbf{0} & \mathbf{I} \end{bmatrix} > 0, \tag{6.21}$$

$$\begin{bmatrix} \bar{\mathbf{W}}_j & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}_{2z,j}^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{D}}_{2u,j}^T & \bar{\mathbf{G}}_i + \bar{\mathbf{G}}_i^T - \bar{\mathbf{P}}_{2,j} \end{bmatrix} > 0, \tag{6.22}$$

and for $j = 1, \cdots, M-1$, *and* $\ell = j+1, \cdots, M$,

$$\begin{bmatrix} \mathbf{\Theta}_1 & * & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_\ell^T + \bar{\mathbf{G}}_i^T \bar{\mathbf{A}}_j^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{B}}_{u\ell}^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{B}}_{uj}^T & \mathbf{\Theta}_2 & * \\ \bar{\mathbf{B}}_{2\omega,j}^T + \bar{\mathbf{B}}_{2\omega,\ell}^T & \mathbf{0} & 2\mathbf{I} \end{bmatrix} > 0, \tag{6.23}$$

$$\begin{bmatrix} \bar{\mathbf{W}}_j + \bar{\mathbf{W}}_\ell & * \\ \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}_{2z,j}^T + \bar{\mathbf{G}}_i^T \bar{\mathbf{C}}_{2z,\ell}^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{D}}_{2u,j}^T + \bar{\mathbf{X}}_i^T \bar{\mathbf{D}}_{2u,\ell}^T & \mathbf{\Theta}_2 \end{bmatrix} > 0, \tag{6.24}$$

and

$$\bar{Z} - \bar{\mathbf{C}}_{2z,i}\, \mathbf{P}_{2,i}\, \bar{\mathbf{C}}_{2z,i}^T \geq 0, \quad \forall i = 1, \cdots, n_\alpha \tag{6.25}$$

where,

$$\begin{aligned} \bar{\mathbf{P}}_{2,j} &= \sum_{i=1}^{n_\theta} f_i^j \mathbf{P}_{2,i}, & \bar{\mathbf{W}}_j &= \sum_{i=1}^{n_\theta} f_i^j \mathbf{W}_i, & \widetilde{\mathbf{P}}_{2,j} &= \sum_{i=1}^{n_\theta} \left(f_i^j + h_i^j\right) \mathbf{P}_{2,i}, \\ \bar{\mathbf{G}}_j &= \sum_{i=1}^{n_\theta} f_i^j \mathbf{G}_i, & \bar{\mathbf{X}}_j &= \sum_{i=1}^{n_\theta} f_i^j \mathbf{X}_i, & \bar{\mathbf{W}}_\ell &= \sum_{i=1}^{n_\theta} f_i^\ell \mathbf{W}_i \end{aligned}$$

$$\begin{aligned} \mathbf{\Theta}_1 &= \sum_{i=1}^{n_\theta} \left(f_i^j + h_i^j\right) \mathbf{P}_{2,i} + \sum_{i=1}^{n_\theta} \left(f_i^\ell + h_i^\ell\right) \mathbf{P}_{2,i}, \\ \mathbf{\Theta}_2 &= 2\bar{\mathbf{G}}_i + 2\bar{\mathbf{G}}_i^T - \sum_{i=1}^{n_\theta} f_i^j \mathbf{P}_{2,i} - \sum_{i=1}^{n_\theta} f_i^\ell \mathbf{P}_{2,i}, \end{aligned}$$

Then, the parameter-dependent noisy static output-feedback gain is given by

$$\begin{aligned} \mathbf{K}(\alpha(k)) &= \hat{\mathbf{X}}(\alpha(k))\, \hat{\mathbf{G}}(\alpha(k))^{-1}, \\ \hat{\mathbf{X}}(\alpha(k)) = \sum_{i=1}^{n_\alpha} \alpha_i(k)\, \mathbf{X}_i, &\quad \hat{\mathbf{G}}(\alpha(k)) = \sum_{i=1}^{n_\alpha} \alpha_i(k)\, \mathbf{G}_i, \end{aligned} \tag{6.26}$$

that exponentially stabilizes the system (6.12) *and satisfies output constraint* (6.14) *with minimal* *control energy given by*

$$\begin{aligned}\mathcal{U}_{OCC} &= trace\Big(\mathbf{K}(\alpha(k))\widetilde{\mathbf{C}}_y\mathbf{P}_{\mathbf{G}}(\alpha(k))\widetilde{\mathbf{C}}_y^T\mathbf{K}^T(\alpha(k)\Big)\\ &\leq \min_{\mathbf{P}_{2,i},\mathbf{W}_i,\mathbf{G}_i,\mathbf{Z}_i}\ \max_i\ trace\big(\mathbf{W}_i\big) = \bar{U},\end{aligned} \tag{6.27}$$

Proof 30 *Due to space limitation, the proof of Lemma 29 is omitted; see [White et al., 2016] for details of a similar proof.*

6.4 Simulation Results and Discussion

The numerical example presented in this section is to demonstrate the advantage of considering sensor performance degradation due to aging during the control design, where the regulated output is subject to the given constraint. The synthesized controller is gain-scheduled by a combination of the system time-varying parameters and the estimated time-varying sensor measurement noise covariance. Note that, the control effort is minimized subject to constraint on the regulated output covariance, in the presence of gradual sensor performance degradation due to sensor aging. Therefore, the sensor aging effect on the system performances is also addressed.

Example: consider a discrete-time multiple-input multiple-output (MIMO) LPV system, originally used in [De Oliveira et al., 1999] and later used in [White et al., 2016] and [Apkarian et al., 2000], with the following state-space matrices

$$\begin{cases} \mathbf{A}(\alpha(k)) = \begin{bmatrix} -2+\alpha_1(k)+\alpha_2(k) & 0 & -1 \\ 1 & -0.5 & 0 \\ 0 & 1 & -0.5 \end{bmatrix}, \mathbf{B}_u(\theta(k)) = \begin{bmatrix} 1+\alpha_1(k)+\alpha_2(k) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{B}_{2\omega} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{C}_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{D}_{2u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{C}_{2z} = \mathbf{I}_3, \end{cases} \tag{6.28}$$

where the time-varying parameters are assumed to have the following parameter variation bounds $\alpha_1(k) \in [-0.5, 0.5]$, $\alpha_2(k) \in \left[10^{-7}, 1\right]$. The discrete-time LPV system (6.28) is converted to the discrete-time polytopic one by solving matrices $\mathbf{A}(\alpha(k))$ and $\mathbf{B}(\alpha(k))$ at the vertices of the parameter space polytope of α_1 and α_2, following the procedure in [White et al., 2013c]. As mentioned earlier, to match with the strictly proper requirements for finite $\mathcal{H}_2$ norm (i.e., feasible $\mathcal{H}_2$ control) in this work, a filter is used for the measured output $\mathrm{y}(\cdot)$ and the control law (6.11) uses the filtered output $\widetilde{y}(\cdot)$, where the z-transform of the filtered measured output is given by

$$\widetilde{Y}(z) = G(z)\, Y(z), \tag{6.29}$$

where $Y(z)$ is the z-transform of the measured output $\mathrm{y}(\cdot)$; $\widetilde{Y}(z)$ is the z-transform of the filtered measured output $\widetilde{\mathrm{y}}(\cdot)$; $G(z) = diag(G_i(z))$ is the transfer function matrix of the filter and each transfer function is given by

$$G_i(z) = \frac{1 - e^{-aT}}{1 - e^{-aT} Z^{-1}}, \quad \forall i = 1, \cdots, q, \tag{6.30}$$

where a is the pole of the filter; T is the sampling period; and q denotes the number of measured output vector components. For the current example, the filter parameters is set as $a = 4.65$ and $T = 0.004$. Generally, $\forall i = 1, \cdots, q$,

$$\begin{aligned} \widetilde{\mathrm{y}}_i(k+1) &= (1 - e^{-aT})\,\mathbf{C}_{yi}\,\mathrm{x}(k+1) + e^{-aT}\,\widetilde{\mathrm{y}}_i(k) + (1 - e^{-aT})\,\widetilde{\nu}(k), \\ &= (1 - e^{-aT})\,\mathbf{C}_{yi}\,\mathbf{A}(\cdot)\mathrm{x}(k) + (1 - e^{-aT})\,\mathbf{C}_{yi}\,\mathbf{B}_u(\cdot)u(k) \\ &+ (1 - e^{-aT})\,\mathbf{C}_{yi}\,\mathbf{B}_{2\omega}(\cdot)\omega_2(k) + e^{-aT}\,\widetilde{\mathrm{y}}_i(k) + (1 - e^{-aT})\,\widetilde{\nu}(k+1), \end{aligned} \tag{6.31}$$

where $\widetilde{\mathrm{y}}_i(\cdot)$ is the i^{th} component of the vector $\widetilde{\mathrm{y}}(\cdot)$; and $\mathbf{C}_{yi}$ is the i^{th} raw of the matrix $\mathbf{C}_y$ which has three raws, i.e., $i \in \{1, 2, 3\}$. Together with, define the augmented states $\widetilde{\mathrm{x}}(\cdot) = \left[\mathrm{x}^T(\cdot) \;\; \widetilde{\mathrm{y}}^T(\cdot)\right]^T$, it yields

$$\widetilde{\mathrm{x}}(k+1) = \underbrace{\begin{bmatrix} \mathbf{A}(\cdot) & 0 & 0 & 0 \\ (1-e^{-aT})\,\mathbf{C}_{y1}\mathbf{A}(\cdot) & e^{-aT} & 0 & 0 \\ (1-e^{-aT})\,\mathbf{C}_{y2}\mathbf{A}(\cdot) & 0 & e^{-aT} & 0 \\ (1-e^{-aT})\,\mathbf{C}_{y3}\mathbf{A}(\cdot) & 0 & 0 & e^{-aT} \end{bmatrix}}_{\widetilde{\mathbf{A}}(\cdot)} \underbrace{\begin{bmatrix} \mathrm{x}^T(k) \\ \\ \widetilde{\mathrm{y}}^T(k) \end{bmatrix}}_{\widetilde{\mathrm{x}}(k)} + \underbrace{\begin{bmatrix} \mathbf{B}_u(\cdot) \\ \\ (1-e^{-aT})\,\mathbf{C}_{y1}\,\mathbf{B}_u(\cdot) \\ (1-e^{-aT})\,\mathbf{C}_{y2}\,\mathbf{B}_u(\cdot) \\ (1-e^{-aT})\,\mathbf{C}_{y3}\,\mathbf{B}_u(\cdot) \end{bmatrix}}_{\widetilde{\mathbf{B}}_{2\omega}(\cdot)} \mathrm{u}(k)$$

$$+ \underbrace{\begin{bmatrix} \mathbf{B}_{2\omega}(\cdot) & \mathbf{0} \\ (1-e^{-aT})\mathbf{C}_{y1}\mathbf{B}_{2\omega}(\cdot) & \\ (1-e^{-aT})\,\mathbf{C}_{y2}\,\mathbf{B}_u(\cdot) & (\mathbf{1}-\mathbf{e}^{-\mathbf{aT}}) \\ (1-e^{-aT})\mathbf{C}_{y3}\mathbf{B}_{2\omega}(\cdot) & \end{bmatrix}}_{\widetilde{\mathbf{B}}_{\omega}(\cdot)} \underbrace{\begin{bmatrix} \omega_2^T(k) \\ \widetilde{\nu}^T(k), \end{bmatrix}}_{\widetilde{\omega}_p(\cdot)}$$

$$\widetilde{\mathrm{z}}_2(k) = \begin{bmatrix} \mathbf{C}_{2_z}(\cdot) & \mathbf{0} \end{bmatrix} \widetilde{\mathrm{x}}(k) + \widetilde{\mathbf{D}}_{2u}(\alpha(k))\,\mathrm{u}(k),$$

$$\widetilde{\mathrm{y}}(k) = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}_y \end{bmatrix} \widetilde{\mathrm{x}}(k), \tag{6.32}$$

where $\mathrm{x}^T(\cdot) = [\mathrm{x}_1(\cdot)\mathrm{x}_2(\cdot)\mathrm{x}_3(\cdot)]^T \in \mathbb{R}^3$ is the states vector; $\widetilde{\mathrm{y}}^T(\cdot) = \left[\widetilde{\mathrm{y}}_1(\cdot)\widetilde{\mathrm{y}}_2(\cdot)\widetilde{\mathrm{y}}_3(\cdot)\right]^T \in \mathbb{R}^3$ is the filtered measured output vector; $\mathbf{1}-\mathbf{e}^{-\mathbf{aT}} = \mathrm{diag}(1-e^{-aT}) \in \mathbb{R}^3$; $\mathbf{0} = \left[0\,0\,0\right]^T \in \mathbb{R}^3$; and $\mathbf{C}_{yi}$ $(i = 1, 2, 3)$ is the i^{th} raw of the matrix $\mathbf{C}_y$.

In the simulation, the output covariance constraint is set to $\mathrm{Cov}(\mathrm{z}(k)) \leq \bar{Z} = \mathscr{Z} \times \mathbf{I}_3$, where $\mathscr{Z}$ is a scalar and $\bar{Z}$ denotes a (3×3) upper bound matrix of output covariance corresponding to the all performance outputs grouped together. The results in Lemma 29 are used with SDP solver SeDuMi [Sturm, 1999] interfaced by the parser YALMIP [Lofberg, 2004] to solve this problem.

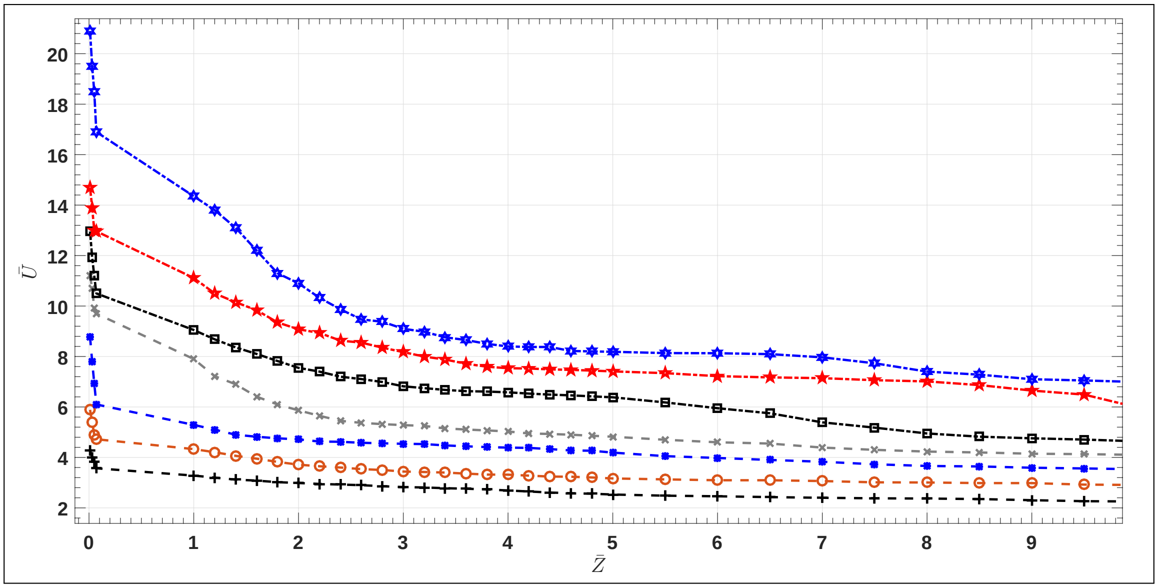


Figure 6.1: Control effort $\bar{U}$ as a function of the output covariance constraints $\bar{Z}$ with different sensor noise covariances for the designed controllers using Lemma. 29 and considering system (5.57). Each line and marks denote the relation between $\bar{U}$ and $\bar{Z}$ at specific noise covariance, for instance, the dotted line with cross marks associated with 10^{-5} noise covariance and dotted line with cross marks associated with 0.9 one. The x-axis is linear in output covariance constraint and y-axis is linear for control effort.

Fig. 6.1 shows the upper bound control effort $\bar{U}$ as a function of the upper bound output covariance constraint $\bar{Z} = \mathscr{Z} \times \mathbf{I}_3$ with different sensor noise covariances for the designed controllers using Lemma 29 and considering system (6.28). Each line and marks denote the relation between $\bar{U}$ and $\bar{Z}$ at a specific noise covariance, for instance, the dotted-line with cross marks associated with 10^{-5} noise covariance and dotted-line with cross marks associated with 0.9 one. The x-axis is linear in output covariance constraint and the y-axis is linear for control effort. This figure has an interesting interpretation: for a given output performance (output covariance), the control energy level (i.e., actuator limit) in terms of OCC cost can be found under sensor aging effect. In other words, for a given actuator limit and under sensor aging effect, the optimal output performance (output covariance) can be found. In addition, as the output covariance constraint is relatively tight, the required control energy is increased. Moreover, as the sensor gets aged (i.e., sensor noise covariance increase), the required control energy to satisfy the output performance is increased.

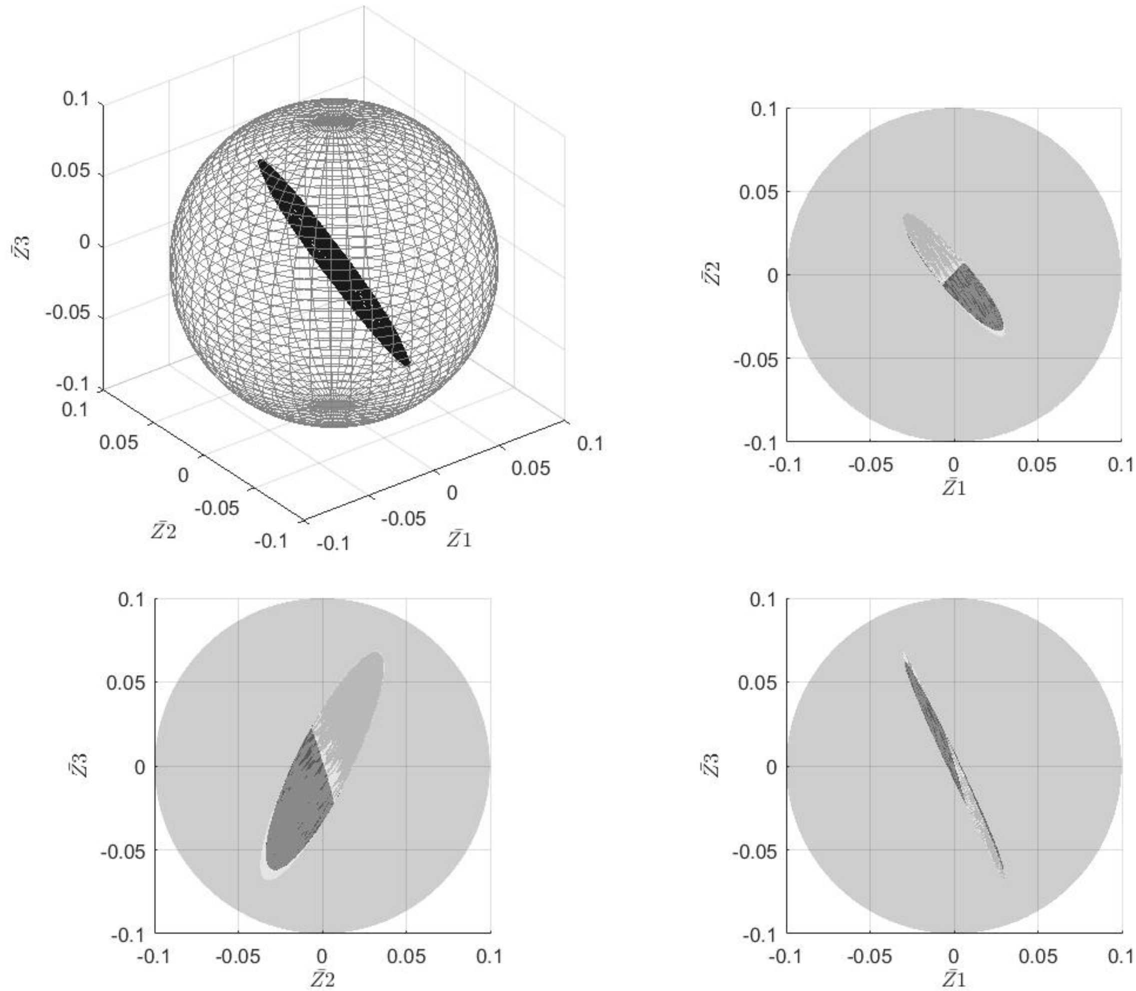
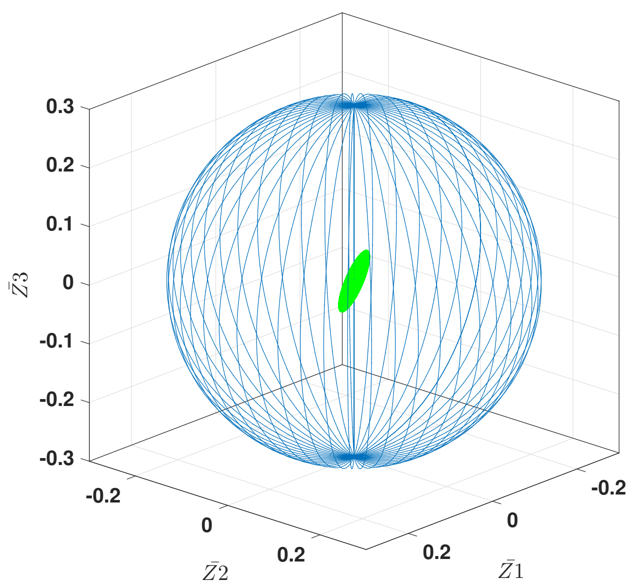
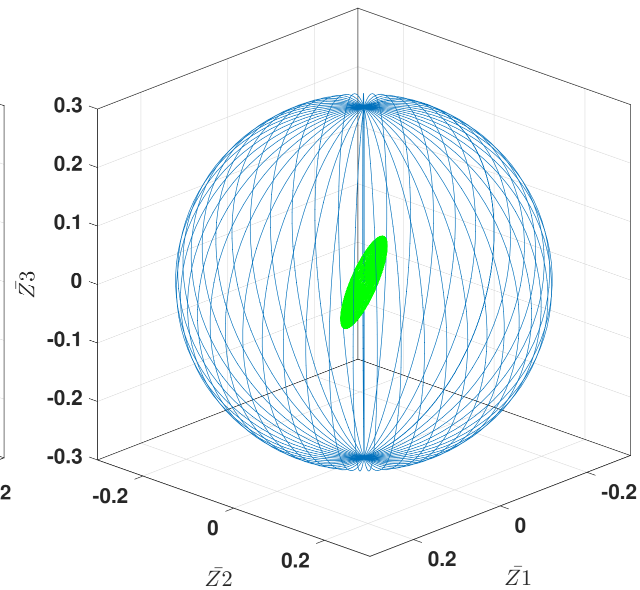


Figure 6.2: The achieved output covariance compared to the designed one, where the sensor noise covariance is set as $\hat{\mathbf{V}} = 10^{-5}$ and designed output covariance ($\bar{Z} = \mathscr{X} \times \mathbf{I}_3$) is set as $\mathscr{X} = 0.1$.

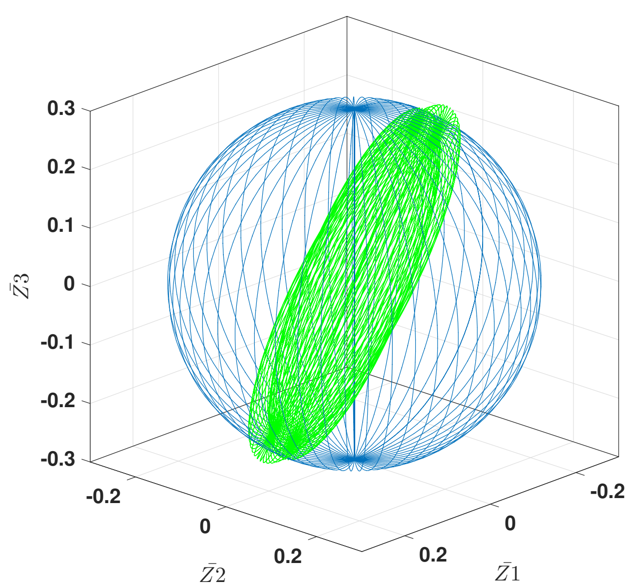
Fig. 6.2 shows the achieved output covariance Cov(z(k)) in (6.14) compared to the design bound $\bar{Z}$. It shows that the designed controller is feasible and the achieved covariance bound is relatively tight with the designed bound in all dimensions. Fig. 6.3 shows the achieved output covariance compared to the designed one for different sensor noise covariance. This figure has an interesting interpretation: as the sensor noise covariance increases (i.e., sensor get aged), achieved output performance approaches the designed one (Fig. 6.3a, 6.3b) and exceeds it (Fig. 6.3c). Then, as the sensor gets aged the designed output covariance needs to be relaxed (Fig. 6.3d) or the sensor needs to be replaced to achieve the required performance.



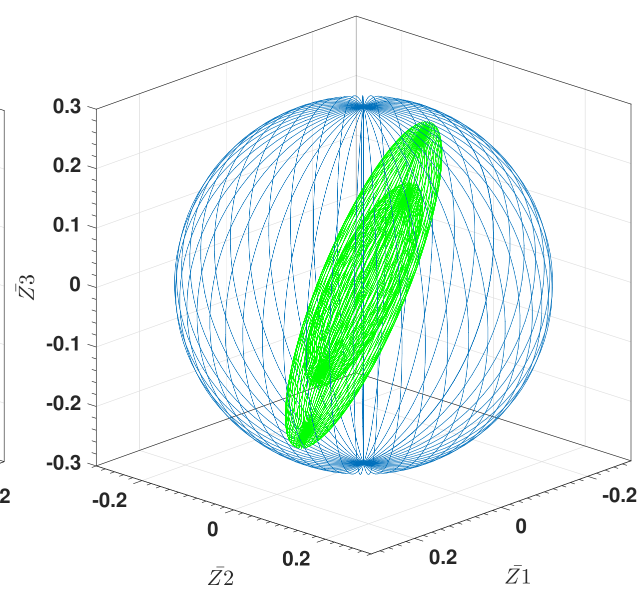
(a)



(b)



(c)



(d)

Figure 6.3: Achieved output covariance compared to the designed one ($\bar{Z} = \mathscr{Z} \times \mathbf{I}_3$), for different estimated sensor noise covariances $\hat{\mathbf{V}}$, where (a) $\hat{\mathbf{V}} = 10^{-5}$, $\mathscr{Z} = 0.3$, (b) $\hat{\mathbf{V}} = 10^{-2}$, $\mathscr{Z} = 0.3$, (c) $\hat{\mathbf{V}} = 0.1$, $\mathscr{Z} = 0.3$, (d) $\hat{\mathbf{V}} = 0.1$, $\mathscr{Z} = 0.301$.

6.5 Summary

The output covariance constraints (OCC) control problem is an optimal control problem that minimizes the control effort subject to performance constraint on the output covariance. Considering that a faulty sensor may lead to degraded system performance, system instability, or even a fatal accident. This work presents the characterization of the OCC control synthesis conditions using linear matrix inequalities (LMI) for a gain-scheduled noisy output-feedback controller that minimizes the cost on control effort with satisfactory system output covariance (OCC control) in the presence of sensor aging. The main motivation is to show the advantage of utilizing sensor performance degradation information in gain-scheduled control. Note that for this work, the gain-scheduled controller is a function of both system time-varying parameters and the estimated sensor measurement noise covariance (varying as sensors age). The proposed controller is validated in a simulation study for a discrete-time linear parameter varying (LPV) system with gradually performance degraded sensors. Simulation results show that the designed controller satisfy the required performance when the sensor performance degradation information is considered. In addition, as the sensor gets further aged the output covariance constraint needs to be relaxed to have a feasible solution. Moreover, if the output performance constraint cannot be relaxed, aging sensor(s) need to be replaced to achieve the desired output covariance performance.

Chapter 7

Conclusions and Future Research

The control systems used in industry and other fields should be safe, reliable, and stable. Degradation of sensor performance due to the sensor aging can be a crucial factor affecting these system performances, reliability, and even stability. Indeed, control system performance is heavily dependent on the sensor signals used for feedback control. That is, sensor performance degradation or fault affects the system output performance directly and could lead to degraded system performance, an unstable system and/or even a fatal accident. Then, adapting the control system to sensor performance variations helps to avoid any catastrophic consequences. This problem motivates this work for monitoring sensor performance and detecting the gradual and/or sudden sensor performance degradation due to sensor aging. In addition, the designed gain-scheduled control for linear parameter varying systems utilizes the sensor performance degradation due to its aging to compensates for sensor aging and maintains the desired performance, where the control is subjected to given constraints. This chapter summarizes concluding remarks and suggestions for future research directions.

7.1 Conclusions

In the following, the main points conclude of this work:

- In contrast to recent fault detection literature, in this work, the sensor performance shift (i.e., sensor performance gradual-degradation and/or sudden fault due to its aging) can be characterized by sensor measurement noise covariance shift. That is, the gradual-degradation of sensor performance due to sensor aging is characterized by the slowly-varying sensor measurement noise covariance, and the sensor abrupt (or intermittent) failures are characterized by an abrupt (or intermittent) change of sensor measurement noise covariance.

To be specific, in contrast to recent literature, model the sensor fault due to aging as stationary augmented state, this work assumes that the sensor performance shift (i.e., sensor performance gradual-degradation and/or fault due to sensor aging) can be characterized by its measurement noise covariance shift, comparing with the most of the noise statistics identification literature for the case with time-invariant sensor noise covariance only.

- A new algorithm is proposed to detect gradual sensor performance degradation due to sensor aging that is characterized by slowly-varying sensor measurement noise covariance.

  To be specific, this work incorporates the innovation vector sequence quality information with the weighted measurements, for estimating the slowly-varying sensor measurement noise covariance. That is, the proposed algorithm invests adaptive filtering theory with the covariance-matching technique and the information about the quality of the weighted innovation vector sequence for the estimation process. Moreover, the use of weighted information of the innovation vector sequence reduces computational load, and the exponentially weighted estimation window improves the estimation accuracy and makes it feasible for online applications. Then, the proposed algorithm is suitable for online applications due to the low computation load and iterative manner; and is less sensitive to initial conditions (i.e., less sensitive to initial state and its initial estimation error covariance matrix). The proposed algorithm to detect gradual sensor performance degradation due to sensor aging shows a reasonable rate of convergence, improved estimation accuracy, and reduced computation load.

- An algorithm is proposed to detect and identify the sudden (abrupt or intermittent) sensor failure, assuming that the sensor performance sudden shift can be characterized by the sudden noise covariance variation of the sensor measurement.

  To be specific, a memory-based technique for detecting the sudden change of sensor measurement noise covariance is proposed. The proposed change detection technique calculates the change of the estimated sensor measurement noise covariance, assuming that the sensor failure is characterized by abruptly (or intermittently) changing of the estimated sensor measurement noise covariance. That is, the change is quantified by the distance between the sensor measurement noise covariance matrices over the two estimation windows before and after the change. The proposed algorithm to detect and identify the abrupt or intermittent sensor failure is developed for LTV systems and applied to LPV systems as a special case. That is, it estimates the slow-varying sensor noise covariance under different covariance variation functions and different scheduling parameter uncertainties, and detects the noise covariance under different abrupt or intermittent change scenarios when the estimation window length is properly selected. The iterative nature of the proposed algorithm significantly reduces the computational load and make it feasible for real-time applications.

- A multi-objective gain-scheduled control for linear parameter varying systems utilizes the sensor performance degradation due to sensor aging, where the controller is subjected to constraints on the control input variances and constraint on the $\mathcal{H}_\infty$ performance output, is designed using the LPV technique.

  To be specific, the synthesis of multi-objective gain-scheduled control is formulated as mixed

ICC/$\mathcal{H}_\infty$ control problem that minimizes the performance output upper bound subject to $\mathcal{H}_2$ constraints on control input variances and $\mathcal{H}_\infty$ performance output. Considering this control, three kinds of controllers are designed and they are: controller scheduled by a combination of the time-varying parameters and the time-varying noise covariance (called fully gain-scheduled), controller scheduled by the sensor noise covariance (called partially gain-scheduled), and not scheduled (called robust controller). This control synthesis problem can be solved efficiently using LMI optimization techniques, where the proposed controller utilizes the sensor fault information (in terms of sensor noise covariance) to minimize its effect on the system and improves the closed-loop system performance. The synthesized multi-objective gain-scheduled noisy output-feedback (GSNOF) controller compensates for the gradually-degraded sensor performance due to sensor aging and maintains the desired performance. The simulation results show that the synthesized fully gain-scheduled controller performs better than the partially gain-scheduled and robust one in terms of the performance with gradually-degraded sensor performance. In addition, the gain-scheduled controllers' performance approaches the robust one when the sensor noise covariance increases significantly.

- A multi-objective gain-scheduled noisy output-feedback controller for linear parameter varying systems to minimize the control effort is designed using the LPV technique, where the controller utilizes the sensor performance degradation due to its aging and subjects to performance constraint on the output covariance.

  To be specific, the output covariance constraints (OCC) control problem, which is an optimal control problem that minimizes the control effort subject to performance constraint on the output covariance, is considered. The characterization of the OCC control synthesis conditions is presented using linear matrix inequalities (LMI) for a gain-scheduled noisy output-feedback controller that minimizes the cost on control effort with satisfactory system output covariance (OCC control) in the presence of sensor aging. The advantage of utilizing sensor performance degradation information in synthesis the gain-scheduled OCC control is demonstrated, where the gain-scheduled controller is a function of both system time-varying parameters and the estimated sensor measurement noise covariance. The proposed controller for the OCC control problem is validated in a simulation study for a discrete-time linear parameter varying (DLPV) system with gradually performance degraded sensors. The designed controller shows ability to satisfy the required performance when the sensor performance degradation information is considered. In addition, as the sensor gets further aged, the output covariance constraint needs to be relaxed to have a feasible solution. Moreover, if the output performance constraint cannot be relaxed, the aging sensor(s) need to be replaced to achieve the desired output covariance performance.

Therefore, algorithms for sensor performance degradation or sensor fault detection and identification due to sensor aging are presented. A gain-scheduled controls for linear parameter varying systems utilizes the sensor performance degradation due to sensor aging are synthesized, where the controller is subjected to multiple constraints. In addition, the importance of considering senor aging information on system stability and performance is demonstrated, where the proposed controller is subject to multiple constraints. Moreover, with the proposed algorithms and designed controllers and considering the desired performance or the available control effort, a decision of accepting or discarding aging sensor measurements can be made.

7.2 Recommendations for Future Research

This section presents recommended directions for future research.

- Implement the proposed algorithms for sensor gradual performance degradation and sensor performance sudden (abrupt or intermittent) change, where sensor performance shift is characterized by sensor measurement noise shift.

- Implement the gain-scheduled controls for linear parameter varying systems utilizes the sensor performance degradation due to sensor aging, where the controller is subjected to multiple constraints (i.e., ICC or OCC control problem).

APPENDICES

Appendix A

Weighted Innovation Vector Estimate

Consider the estimation window with size N, the innovation vector over the estimation window is $\tilde{\mathbf{v}} = [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_{N-1}, \tilde{\mathbf{v}}_N]$, where $\tilde{\mathbf{v}}_1$ is the first (oldest) innovation data collect within the estimation window and $\tilde{\mathbf{v}}_N$ is the last (recent) one. To obtain a feasible estimation based on the available measurements (minimal available information) for reducing estimation error and computational load, the innovation vector need to be weighted such that more weight assigned to more recent innovation data and less weight to more old ones. The weight coefficient $\varphi < 1$ is used as a "forgetting factor" for the past innovation vector data. In this work, the weight is assigned over the estimation window rather than the whole horizon, such that assigned highest weight to $\tilde{\mathbf{v}}_N$ (more recent innovation data collected) and less weight to $\tilde{\mathbf{v}}_1$ (oldest innovation data). that is

$$
\begin{aligned}
\mathbf{v} &= \left[\varphi^{N-1}\, \tilde{\mathbf{v}}_1, \varphi^{N-2}\, \tilde{\mathbf{v}}_2, \cdots, \varphi^{1}\, \tilde{\mathbf{v}}_{N-1}, \tilde{\mathbf{v}}_N\right] \\
&= \left[\varphi^{N-1}\, \tilde{\mathbf{v}}_1, \varphi^{N-2}\, \tilde{\mathbf{v}}_2, \cdots, \varphi^{N-(N-1)}\, \tilde{\mathbf{v}}_{N-1}, \varphi^{N-(N)}\, \tilde{\mathbf{v}}_N\right] \\
&= \left[\varphi^{N-k}\, \tilde{\mathbf{v}}_k\right], \forall k = 1, \cdots, N.
\end{aligned}
$$

Then, the innovation vector estimate is given by

$$
\bar{\mathbf{v}}_N = \frac{1}{N}\left[\sum_{k=1}^{N} \varphi^{N-k}\, \tilde{\mathbf{v}}_k\right],
$$

and that shows Eqn. (3.11).

Appendix B

$\mathcal{H}_2$ Performance

Considering system (5.19) and suppose that system (5.22) is exponentially stable. The definition of the infinite horizon $\mathcal{H}_2$ performance for the LPV system (5.22) is given by

$$\begin{aligned} ||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 &= \lim_{\mathcal{T}\to\infty} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} \mathbb{E}\left\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}} \widetilde{z}_2^T(k)\widetilde{z}_2(k)\right\}, \\ &= \lim_{\mathcal{T}\to\infty} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} \left\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}} trace\left[\left(\mathbf{C}_{2cl}(\alpha(k))\,\mathbf{P_G}(\alpha(k))\,\mathbf{C}_{2cl}(\alpha(k))^T\right)\right]\right\}, \end{aligned}$$

where Λ_{2n_θ} denotes the polytopic space, where the scheduling parameters belong to; $\mathbb{E}$ denotes the expectation operator; the positive integer $\mathcal{T}$ denotes the time horizon; and $\mathbf{P_G}(\alpha(k))$ is the solution of the discrete-time Lyapunov equation (5.29), which is restated below for convenient.

$$\mathbf{P_G}(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P_G}(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\,\mathbf{B}_{2cl}^T(\alpha(k)),\ \mathbf{P_G}(0) = 0.$$

Below are the steps followed to reach result presented above. Note that, the logical structure of the derivation below are borrowed from Ref [De Caigny et al., 2010] with slight adjustment to match this work direction. First, the finite horizon is considered, and the extension to infinite horizon will be straight forward. Let the finite horizon is $k \in \{0, \mathcal{T}\}$ and recall that the $\widetilde{\omega}_{2p}(k)$ is white noise with unity covariance matrix, then, the output $\widetilde{z}_2$ in system (5.22) can be calculated as

$$\begin{aligned}\widetilde{z}_2 &= \mathbf{C}_{2cl}(\alpha(k))\Bigg(\sum_{j=0}^{k-2}\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)\mathbf{B}_{2cl}(\alpha(j))\,\widetilde{\omega}_{2p}(j)+\mathbf{B}_{2cl}(\alpha(k-1))\,\widetilde{\omega}_{2p}(k-1)\Bigg),\\ &= \sum_{j=0}^{k}\mathcal{A}(k,j)\widetilde{\omega}_{2p}(j),\end{aligned}$$

where $\mathcal{A}(k,j) = \mathbf{C}_{2cl}(\alpha(k))\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\mathbf{B}_{2cl}(\alpha(j))\Big)$, with $\prod_{i=k}^{k-1}\mathbf{A}_{cl}(\alpha(i)) = \mathbf{I}$ if $j \neq k$, and 0 otherwise. Considering the mathematical identity

$$\widetilde{z}_2^T(k)\,\widetilde{z}_2(k) = trace\big[\widetilde{z}_2(k)\,\widetilde{z}_2^T(k)\big],$$

it yields

$$\begin{aligned}||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 &= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\mathbb{E}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}\Big(trace\big[\widetilde{z}_2(k)\,\widetilde{z}_2^T(k)\big]\Big)\Bigg\},\\
&= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\mathbb{E}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}trace\Big[\Big(\sum_{j=0}^{k}\mathcal{A}(k,j)\widetilde{\omega}_{2p}(j)\Big)\Big(\sum_{\ell=0}^{k}\mathcal{A}(k,\ell)\widetilde{\omega}_{2p}(\ell)\Big)^T\Big]\Bigg\},\\
&= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}trace\Big[\Big(\sum_{j=0}^{k}\sum_{\ell=0}^{k}\mathcal{A}(k,j)\,\mathbb{E}\Big(\widetilde{\omega}_{2p}(j)\widetilde{\omega}_{2p}(\ell)^T\Big)\mathcal{A}(k,\ell)^T\Big)\Big]\Bigg\},\\
&= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}trace\Big[\sum_{j=0}^{k}\mathcal{A}(k,j)\mathcal{A}(k,j)^T\Big]\Bigg\},\\
&= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}trace\Big[\sum_{j=0}^{k-1}\mathbf{C}_{2cl}(\alpha(k))\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)\mathbf{B}_{2cl}(\alpha(j))\\
&\quad\times\mathbf{B}_{2cl}(\alpha(j))^T\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)^T\mathbf{C}_{2cl}(\alpha(k))^T\Big]\Bigg\},\\
&= \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\Bigg\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}}trace\big[\big(\mathbf{C}_{2cl}(\alpha(k))\,\mathbf{P}_{\mathbf{G}}(\alpha(k))\,\mathbf{C}_{2cl}(\alpha(k))^T\big)\big]\Bigg\}\end{aligned}$$

where

$$\mathbf{P_G}(\alpha(k)) = \sum_{j=0}^{k-1}\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)\mathbf{B}_{2cl}(\alpha(j))\mathbf{B}_{2cl}(\alpha(j))^T\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)^T, \text{with } \mathbf{P_G}(0) = 0,$$

then

$$\begin{aligned}\mathbf{P_G}(\alpha(k+1)) &= \sum_{j=0}^{k}\Big(\prod_{i=j+1}^{k}\mathbf{A}_{cl}(\alpha(i))\Big)\mathbf{B}_{2cl}(\alpha(j))\mathbf{B}_{2cl}(\alpha(j))^T\Big(\prod_{i=j+1}^{k}\mathbf{A}_{cl}(\alpha(i))\Big)^T,\\ &= \mathbf{A}_{cl}(\alpha(k))\sum_{j=0}^{k-1}\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)\mathbf{B}_{2cl}(\alpha(j))\mathbf{B}_{2cl}(\alpha(j))^T\Big(\prod_{i=j+1}^{k-1}\mathbf{A}_{cl}(\alpha(i))\Big)^T\mathbf{A}_{cl}(\alpha(k))^T\\ &\quad + \mathbf{B}_{2cl}(\alpha(k))\mathbf{B}_{2cl}(\alpha(k))^T,\\ &= \mathbf{A}_{cl}(\alpha(k))\mathbf{P_G}(\alpha(k))\mathbf{A}_{cl}(\alpha(k))^T + \mathbf{B}_{2cl}(\alpha(k))\mathbf{B}_{2cl}(\alpha(k))^T,\end{aligned}$$

Then, the finite horizon $\mathcal{H}_2$ performance is given by

$$||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 = \sup_{\alpha(k)\in\Lambda_{2n_\theta}}\left\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}} trace\big[\big(\mathbf{C}_{2cl}(\alpha(k))\,\mathbf{P_G}(\alpha(k))\,\mathbf{C}_{2cl}(\alpha(k))^T\big)\big]\right\}$$

where $\mathbf{P_G}(\alpha(k))$ satisfy

$$\mathbf{P_G}(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P_G}(\alpha(k))\mathbf{A}_{cl}(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\mathbf{B}_{2cl}(\alpha(k))^T, \mathbf{P_G}(0) = 0,$$

Then the infinite horizon of the $\mathcal{H}_2$ performance is proved. Since the infinite horizon is obtained by let $\mathcal{T} \to \infty$ of the finite horizon is $k \in \{0, \mathcal{T}\}$ Therefore, the infinite horizon $\mathcal{H}_2$ performance is is given by

$$||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 = \lim_{\mathcal{T}\to\infty}\sup_{\alpha(k)\in\Lambda_{2n_\theta}}\left\{\frac{1}{\mathcal{T}}\sum_{k=0}^{\mathcal{T}} trace\big[\big(\mathbf{C}_{2cl}(\alpha(k))\,\mathbf{P_G}(\alpha(k))\,\mathbf{C}_{2cl}(\alpha(k))^T\big)\big]\right\} \qquad (\star)$$

where $\mathbf{P_G}(\alpha(k))$ defined above.

Appendix C

Proof of Lemma 21

Assume that $\mathbf{A}_{cl}(\alpha(k))$ in Eqn. (5.22) is stable for all $\alpha(k) \in \Lambda_{2n_\theta}$. Then, there is a symmetric positive-definite matrix, $\mathbf{P_G}(\alpha(k)) = \mathbf{P_G}^T(\alpha(k)) > 0$, satisfies

$$\mathbf{P_G}(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P_G}(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\,\mathbf{B}_{2cl}^T(\alpha(k)),$$

Considering the parameter-dependent Lyapunov matrix function is $\mathcal{V}(\alpha(k)) = x_k^T\,\mathbf{P}_2(\alpha(k))\,x_k$, substituting it in discrete-time Lyapunov equation above, it yields

$$-\mathbf{P}_2(\alpha(k+1)) + \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\,\mathbf{B}_{2cl}^T(\alpha(k)) < 0.$$

Applying the Schur complement to above inequality, it leads

$$\begin{bmatrix} \mathbf{P}_2(\alpha(k+1)) - \mathbf{A}_{cl}^T(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) & * \\ \mathbf{B}_{2cl}^T(\alpha(k)) & \mathbf{I} \end{bmatrix} > 0,$$

and to ensure that last LMI is satisfied under assumption $\mathbf{A}_{cl}(\cdot)$ is stable, it is required that $\mathbf{P}_2(\cdot) = \mathbf{P}_2^T(\cdot) > 0$. Last inequality could be re-written as

$$\begin{bmatrix} \mathbf{P}_2(\alpha(k+1)) & \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_2(\alpha(k)) & \mathbf{B}_{2cl}(\alpha(k)) \\ \mathbf{P}_2(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) & \mathbf{P}_2(\alpha(k)) & \mathbf{0} \\ \mathbf{B}_{2cl}^T(\alpha(k)) & \mathbf{0} & \mathbf{I} \end{bmatrix} > 0,$$

Considering the change of variable $\mathbf{K}(\alpha(k)) = \mathbf{Z}(\alpha(k))\ \mathbf{G}(\alpha(k))^{-1}$, and multiply above inequality from left by $daig\left(\begin{bmatrix}\mathbf{I} & \mathbf{G}(\alpha(k))^{-1}\mathbf{P}(\alpha(k)) & \mathbf{I}\end{bmatrix}\right)$ and $daig\left(\begin{bmatrix}\mathbf{I} & \mathbf{G}(\alpha(k))^{-1}\mathbf{P}(\alpha(k)) & \mathbf{I}\end{bmatrix}^T\right)$ from right, it yields the LMI (5.30). Additionally, since

$$\mathbf{P}_2(\alpha(k+1)) - \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) - \mathbf{B}_{2cl}(\alpha(k))\ \mathbf{B}_{2cl}^T(\alpha(k)) > 0,$$

there exist matrix $\mathbf{M}(\cdot) = \mathbf{M}^T(\cdot)$ such that

$$\mathbf{P}_2(\alpha(k+1)) = \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) + \mathbf{B}_{2cl}(\alpha(k))\ \mathbf{B}_{2cl}^T(\alpha(k)) + \mathbf{M}(\alpha(k)).$$

Therefore, the symmetric positive-definite Lyapunov matrix satisfy

$$\mathbf{P}_2(\alpha(k)) = \mathbf{P}_2^T(\alpha(k)) > \mathbf{P}_{\mathbf{G}}(\alpha(k)) = \mathbf{P}_{\mathbf{G}}{}^T(\alpha(k)) > 0.$$

Consequently,

$$\mathbf{C}_{2cl}(\theta(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k) > \mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_{\mathbf{G}}(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k), \qquad (\star\star)$$

taking trace of both sides and callback Eqn. ($\star$), yields

$$||\mathcal{S}_{\widetilde{z}_2,\widetilde{\omega}_{2p}}(\alpha(k))||_2^2 \leq \lim_{\mathcal{T}\to\infty} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} \left\{ \frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\big[\big(\mathbf{C}_{2cl}(\alpha(k))\, \mathbf{P_2}(\alpha(k))\, \mathbf{C}_{2cl}(\alpha(k))^T\big)\big] \right\}.$$

Multiply LMI (5.31) by $[I \;\; \mathbf{C}_{2cl}]$ from left and by $[I \;\; \mathbf{C}_{2cl}]^T$ from right, it yields

$$\mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k) - \mathbf{W}(\alpha(k)) < 0,$$

$$\mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k) < \mathbf{W}(\alpha(k))$$

then taking trace of both sides, it yields

$$trace\Big(\mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k)\Big) < trace\Big(\mathbf{W}(\alpha(k))\Big).$$

Taking summation over finite horizon of both sides, yields

$$\frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\Big(\mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k)\Big) < \frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\Big(\mathbf{W}(\alpha(k))\Big).$$

Since,

$$\frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\Big(\mathbf{W}(\alpha(k))\Big) \leq \sup_{\alpha(k)\in\Lambda_{2n_\theta}} trace\Big(\mathbf{W}(\alpha(k))\Big).$$

which implies that

$$\inf_{\mathbf{P}_\infty(\alpha(k)),\mathbf{P_2}(\alpha(k)),\mathbf{G}(\alpha(k)),\mathbf{W}(\alpha(k))} \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\Big(\mathbf{W}(\alpha(k))\Big)$$
$$\leq \inf_{\mathbf{P}_\infty(\alpha(k)),\mathbf{P}(\alpha(k)),\mathbf{G}(\alpha(k)),\mathbf{W}(\alpha(k))} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} trace\Big(\mathbf{W}(\alpha(k))\Big).$$

Consequently,

$$\sup_{\alpha(k)\in\Lambda_{2n_\theta}} \lim_{\mathcal{T}\to\infty} \frac{1}{\mathcal{T}} \sum_{k=0}^{\mathcal{T}} trace\Big(\mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_2(\alpha(k))\mathbf{C}_{2cl}^T(\alpha(k)\Big)$$

$$< \inf_{\mathbf{P}(\alpha(k)),\mathbf{G}(\alpha(k)),\mathbf{W}(\alpha(k))} \sup_{\alpha(k)\in\Lambda_{2n_\theta}} trace\Big(\mathbf{W}(\alpha(k))\Big).$$

Then, minimize $trace\Big(\mathbf{W}(\alpha(k))\Big)$ leads to minimize the upper bound of the ICC cost.
Follow the same manner of proofing the LMI (5.31), multiply Eqn. (5.32) from left by $\Big[I\ \mathbf{\Gamma}_r\mathbf{K}(\alpha(k))\widetilde{\mathbf{C}}_y\Big]$ and from right by $\Big[I\ \mathbf{\Gamma}_r\mathbf{K}(\alpha(k))\widetilde{\mathbf{C}}_y\Big]^T$ it yields

$$\mathbf{\Gamma}_r\mathbf{K}(\alpha(k))\widetilde{\mathbf{C}}_y\mathbf{P}_2(\alpha(k))\widetilde{\mathbf{C}}_y^T\mathbf{K}(\alpha(k))^T\mathbf{\Gamma}_r^T < \bar{\mathbf{U}}_r, \quad \forall r = 1,\cdots,m,$$

As a result, the r^{th} control input covariance is upper bounded by $\bar{\mathbf{U}}_r$.

The proof of LMI (5.33) is the following: recall Lemma 3. of Ref. [de Souza et al., 2006], which states the follows, considering system (5.22) and let η be a given scalar. If there exists a bounded matrix $\mathbf{R}(\alpha(k))$ satisfying, $\forall\alpha(k)\in\Lambda_{2n_\theta}$, the following inequality

$$\begin{bmatrix} \mathbf{P}_\infty(\alpha(k+1)) & * & * & * \\ \mathbf{R}^T(\alpha(k))\mathbf{A}_{cl}^T(\alpha(k)) & \mathbf{R}(\alpha(k)) & * & * \\ \mathbf{B}_{\infty cl}^T(\alpha(k)) & \mathbf{0} & \eta\mathbf{I} & * \\ \mathbf{0} & \mathbf{C}_{\infty cl}(\alpha(k))\mathbf{G}(\alpha(k)) & \mathbf{D}_{\infty cl}(\alpha(k)) & \eta\mathbf{I} \end{bmatrix} > 0, \qquad (\star\star\star)$$

then the system (5.22) is exponentially stable and $||\mathcal{CL}||_\infty \le \eta$, where $\mathcal{CL}$ is the transfer function fo the closed-loop system. If the same scalar η and matrix $\mathbf{R}(\alpha(k))$ satisfying last matrix inequality, it satisfies LMI (5.33) (necessary condition). Now, multiply LMI (5.33) from right by

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A}_{cl} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & -\mathbf{C}_{\infty cl} & 0 & \mathbf{I} \end{bmatrix}$$

and from left by above matrix transpose, it implies the following

$$\begin{bmatrix} \Theta_1 & * & * \\ \mathbf{B}^T_{\infty cl}(\alpha(k)) & \eta \mathbf{I} & * \\ -\mathbf{C}_{\infty cl}(\alpha(k))\mathbf{P}_\infty(\alpha(k))\mathbf{A}^T(\alpha(k)) & \mathbf{D}_{\infty cl}(\alpha(k)) & \Theta_2 \end{bmatrix} > 0.$$

where

$\Theta_1 = \mathbf{P}_\infty(\alpha(k+1)) - \mathbf{A}_{cl}(\alpha(k))\mathbf{P}_\infty(\alpha(k))\mathbf{A}^T_{cl}(\alpha(k))$ and

$\Theta_2 = \eta \mathbf{I} - \mathbf{C}_{\infty cl}(\alpha(k))\mathbf{P}_\infty(\alpha(k))\mathbf{C}^T_{\infty cl}(\alpha(k))$. Finally, by Schur's complement, it follows that the latter inequality is equivalent to Eqn. ($\star \star \star$) (sufficient condition). Therefore, the $\mathcal{H}_\infty$ performance is guaranteed and bounded by η when LMI (5.33) is satisfied.

Appendix D

Proof of Lemma 23

For any $\widetilde{\alpha}(k)$ belongs to the multi-simplex Λ_M, multiply (5.41) by $\widetilde{\alpha}_j^2$ and sum for $j = 1, \cdots, M$. Similarly, multiply (5.48) by $\widetilde{\alpha}_j \; \widetilde{\alpha}_\ell$ and sum for $j = 1, \cdots, M-1$ and $\ell = j+1, \cdots, M$, and then add the two expressions, and then consider the change of variables in Eqn. (5.14) and the parameterizations (5.34) to (5.37), that yields to Eqn. (5.30). The rest of the proof similar to proof of Lemma (21). With satisfying conditions of Lemma (21) for all $\alpha(k) \in \Lambda_{2n_\theta}$ with $k \geq 0$, it follows that

$$\sup_{\alpha \in \Lambda_{2n_\theta}} \; trace\,(\mathbf{W}(\alpha(k))) = \max_i \; trace\,(\mathbf{W}_i),$$

therefore,

$$\inf_{\mathbf{P}_\infty(\alpha), \mathbf{P}_2(\alpha), \mathbf{G}(\alpha), \mathbf{Z}(\alpha), \mathbf{W}(\alpha)} \; \sup_{\alpha \in \Lambda_{2n_\theta}} \; trace\,(\mathbf{W}(\alpha(k))) \;\leq\; \min_{\mathbf{P}_{\infty,i}, \mathbf{P}_{2,i}, \mathbf{G}_i, \mathbf{Z}_i, \mathbf{W}_i} \; \max_i trace\,(\mathbf{W}_i)$$

Then, LMI (5.41) is satisfied. The same procedure applied to LMIs (5.43) to (5.51). Therefore, conditions of Lemma (23) are satisfied, and this end the proof.

Appendix E

Control Input Covariance

Consider the multi-objective gain-scheduled noisy static output-feedback controller (GSNOF) is given by

$$u(k) = \mathbf{K}(\alpha(k))\,\widetilde{y}(k),$$

which can be further partitioned using $\mathbf{\Gamma}_r$, which is a given input channel section matrix for the control input r, as follows

$$u(k) = \big[u_1(k), u_2(k), \cdots, u_m(k)\big]^T,$$

where $u_r(k) = \mathbf{\Gamma}_r u(k)$, $\forall r = 1, \cdots, m$; and $\mathbf{K}(\alpha(k))$ is the gain-scheduled control gain. The covariance of the r^{th} control input is given by

$$\begin{aligned}
Cov(u_r(k)) &= \mathbb{E}\big(u_r(k) u_r(k)^T\big), \quad r = 1, \cdots, m,\\
&= \mathbb{E}\Big[\big(\mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{y}(k)\big)\big(\mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{y}(k)\big)^T\Big],\\
&= \mathbb{E}\Big[\big(\mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y\,\widetilde{x}(k)\big)\big(\mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y\,\widetilde{x}(k)\big)^T\Big],\\
&= \mathbb{E}\Big[\big(\mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y\,\widetilde{x}(k)\big)\big(\widetilde{x}(k)^T\,\widetilde{\mathbf{C}}_y^T\,\mathbf{K}(\alpha(k))^T\,\mathbf{\Gamma}_r^T\big)\Big],\\
&= \mathbf{\Gamma}_r \mathbf{K}(\alpha(k))\,\widetilde{\mathbf{C}}_y\,\mathbf{P}_{\mathbf{G}}(\alpha(k))\,\widetilde{\mathbf{C}}_y^T\,\mathbf{K}(\alpha(k))^T\,\mathbf{\Gamma}_r^T,
\end{aligned}$$

where $\mathbf{P}_{\mathbf{G}}(\alpha(k))$ is the solution of the time-varying Lyapunov equation in (5.29). Note that $\mathbf{P}_{\mathbf{G}}(\alpha(k))$ is the state covariance according to definition 4.4.1 in Ref. [Skelton et al., 1998] (with considering there the case of LTV system), for more details about the derivation of $\mathbf{P}_{\mathbf{G}}(\alpha(k))$; see Ch.4 in

[Skelton et al., 1998].

Appendix F

Control Output Covariance

Consider system (5.22), the $\mathcal{H}_2$ output covariance is given by

$$
\begin{aligned}
Cov(\widetilde{\mathrm{z}}_2(k)) &= \mathbb{E}\big[\widetilde{\mathrm{z}}_2(k)\widetilde{\mathrm{z}}_2(k)^T\big], \\
&= \mathbb{E}\big[\big(\mathbf{C}_{2cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k)\big)\big(\mathbf{C}_{2cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k)\big)^T\big], \\
&= \mathbb{E}\big[\big(\mathbf{C}_{2cl}(\alpha(k))\,\widetilde{\mathrm{x}}(k)\big)\big(\widetilde{\mathrm{x}}(k)^T\mathbf{C}_{2cl}(\alpha(k))^T\big)\big], \\
&= \mathbf{C}_{2cl}(\alpha(k))\mathbf{P}_{\mathbf{G}}(\alpha(k))\mathbf{C}_{2cl}(\alpha(k))^T,
\end{aligned}
$$

where $\mathbf{P}_{\mathbf{G}}(\alpha(k))$ is the solution of the time-varying Lyapunov equation in (5.29).

BIBLIOGRAPHY

BIBLIOGRAPHY

[Abbaspour et al., 2017] Abbaspour, A., Aboutalebi, P., Yen, K. K., and Sargolzaei, A. (2017). Neural adaptive observer-based sensor and actuator fault detection in nonlinear systems: Application in uav. *ISA transactions*, 67(1):317–329.

[Abdallah et al., 2018] Abdallah, M. B., Azar, A. T., Guedri, H., Malek, J., and Belmabrouk, H. (2018). Noise-estimation-based anisotropic diffusion approach for retinal blood vessel segmentation. *Neural Computing and Applications*, 29(8):159–180.

[Abdelmalek et al., 2018] Abdelmalek, S., Azar, A. T., and Dib, D. (2018). A novel actuator fault-tolerant control strategy of dfig-based wind turbines using takagi-sugeno multiple models. *International Journal of Control, Automation and Systems*, 16(3):1415–1425.

[Alvergue et al., 2016] Alvergue, L. D., Pandey, A., Gu, G., and Chen, X. (2016). Consensus control for heterogeneous multiagent systems. *SIAM Journal on Control and Optimization*, 54(3):1719–1738.

[Ananthasayanam, 2018] Ananthasayanam, M. R. (2018). A reference recursive recipe for tuning the statistics of the Kalman filter. In Serra, G. L. d. O., editor, *Kalman Filters-Theory for Advanced Applications*, chapter 01. InTech, Rijeka.

[Apkarian et al., 2000] Apkarian, P., Pellanda, P. C., and Tuan, H. D. (2000). Mixed $H2/H_\infty$ multi-channel linear parameter-varying control in discrete time. *Systems & Control Letters*, 41(5):333–346.

[Arogeti et al., 2012] Arogeti, S. A., Wang, D., Low, C. B., and Yu, M. (2012). Fault detection isolation and estimation in a vehicle steering system. *IEEE Transactions on Industrial Electronics*, 59(12):4810–4820.

[Bagherieh and Nagamune, 2014] Bagherieh, O. and Nagamune, R. (2014). Utilization of blade pitch control in low wind speed for floating offshore wind turbines. In *American Control Conference (ACC), 2014*, pages 4354–4359. IEEE.

[Bar-Shalom and Kirubarajan, 2004] Bar-Shalom, Yaakov, X. R. L. and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley& Sons, New York.

[Bernstein, 2005] Bernstein, D. S. (2005). *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*, volume 41. Princeton University Press, Princeton.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press, Cambridge.

[Brockwell and Davis, 2016] Brockwell, P. J. and Davis, R. A. (2016). *Introduction to time series and forecasting*. Springer, New York.

[Cai and Wu, 2010] Cai, X. and Wu, F. (2010). Robust parameter-dependent fault-tolerant control for actuator and sensor faults. *International Journal of Control*, 83(7):1475–1484.

[Cardenas et al., 2008] Cardenas, A. A., Amin, S., and Sastry, S. (2008). Secure control: Towards survivable cyber-physical systems. In *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*, pages 495–500. IEEE.

[Chamseddine and Noura, 2008] Chamseddine, A. and Noura, H. (2008). Control and sensor fault tolerance of vehicle active suspension. *IEEE Transactions on Control Systems Technology*, 16(3):416–433.

[Chen et al., 2017] Chen, Y. M., Tsai, C.-L., and Fang, R.-W. (2017). TDOA/FDOA mobile target localization and tracking with adaptive extended Kalman filter. In *2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 202–206. IEEE.

[Chu, 2015] Chu, W. (2015). *State estimation and coordinated control for distributed electric vehicles*. Springer Berlin Heidelberg, Berlin.

[Chui and Chen, 2008] Chui, C. K. and Chen, G. (2008). *Kalman filtering: with real-time applications*. Springer-Verlag Berlin Heidelberg, Berlin.

[CusidOCusido et al., 2008] CusidOCusido, J., Romeral, L., Ortega, J. A., Rosero, J. A., and Espinosa, A. G. (2008). Fault detection in induction machines using power spectral density in wavelet decomposition. *IEEE Transactions on Industrial Electronics*, 55(2):633–643.

[De Caigny et al., 2012] De Caigny, J., Camino, J. F., Oliveira, R. C., Peres, P. L., and Swevers, J. (2012). Gain-scheduled dynamic output feedback control for discrete-time LPV systems. *International Journal of Robust and Nonlinear Control*, 22(5):535–558.

[De Caigny et al., 2010] De Caigny, J., Camino, J. F., Oliveira, R. C., Peres, P. L. D., and Swevers, J. (2010). Gain-scheduled H_2 and H_∞ control of discrete-time polytopic time-varying systems. *IET control theory & applications*, 4(3):362–380.

[De Oliveira et al., 1999] De Oliveira, M., Geromel, J., and Bernussou, J. (1999). An LMI optimization approach to multiobjective controller design for discrete-time systems. In *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, volume 4, pages 3611–3616. IEEE.

[de Souza et al., 2006] de Souza, C. E., Barbosa, K. A., and Neto, A. T. (2006). Robust H_∞; filtering for discrete-time linear systems with uncertain time-varying parameters. *IEEE Transactions on Signal Processing*, 54(6):2110–2118.

[Devroye et al., 2006] Devroye, N., Mitran, P., and Tarokh, V. (2006). Achievable rates in cognitive radio channels. *IEEE Transactions on Information Theory*, 52(5):1813–1827.

[Edwards et al., 2010] Edwards, C., Lombaerts, T., and Smaili, H. (2010). *Fault Tolerant Flight Control: A Benchmark Challenge*. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, Berlin.

[Enescu et al., 2002] Enescu, M., Sirbu, M., and Koivunen, V. (2002). Recursive estimation of noise statistics in Kalman filter based MIMO equalization. *the proceedings of XXVIIth General Assembly of the International Union of Radio Science (URSI), Maastricht the Netherlands*, pages 17–24.

[Fakoorian and Swindlehurst, 2013] Fakoorian, S. A. A. and Swindlehurst, A. L. (2013). Full rank solutions for the MIMO Gaussian wiretap channel with an average power constraint. *IEEE Transactions on Signal Processing*, 61(10):2620–2631.

[Feng et al., 2013] Feng, W., Wang, Y., Ge, N., Lu, J., and Zhang, J. (2013). Virtual MIMO in multi-cell distributed antenna systems: coordinated transmissions with large-scale csit. *IEEE Journal on Selected Areas in Communications*, 31(10):2067–2081.

[Foo et al., 2013] Foo, G. H. B., Zhang, X., and Vilathgamuwa, D. M. (2013). A sensor fault detection and isolation method in interior permanent-magnet synchronous motor drives based on an extended Kalman filter. *IEEE Transactions on Industrial Electronics*, 60(8):3485–3495.

[Grewal and Andrews, 2001] Grewal, M. and Andrews, A. (2001). *Kalman filtering: theory and practice using MATLAB*. John Wiley& Sons, New York.

[Hajiyev et al., 2015] Hajiyev, C., Soken, H. E., and Vural, S. Y. (2015). *State estimation and control for low-cost unmanned aerial vehicles*. Springer, Berlin Heidelberg, Berlin.

[Heredia et al., 2008] Heredia, G., Ollero, A., Bejar, M., and Mahtani, R. (2008). Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics*, 18(2):90–99.

[Hu et al., 2018] Hu, L., Wang, Z., Han, Q.-L., and Liu, X. (2018). State estimation under false data injection attacks: Security analysis and system protection. *Automatica*, 87(1):176–183.

[Huh et al., 2011] Huh, H., Moon, S.-H., Kim, Y.-T., Lee, I., and Caire, G. (2011). Multi-cell MIMO downlink with cell cooperation and fair scheduling: A large-system limit analysis. *IEEE Transactions on Information Theory*, 57(12):7771–7786.

[Ingimundarson et al., 2009] Ingimundarson, A., Bravo, J. M., Puig, V., Alamo, T., and Guerra, P. (2009). Robust fault detection using zonotope-based set-membership consistency test. *International journal of adaptive control and signal processing*, 23(4):311–330.

[Isermann, 2005] Isermann, R. (2005). Model-based fault-detection and diagnosis-status and applications. *Annual Reviews in control*, 29(1):71–85.

[Kadlec et al., 2011] Kadlec, P., Grbić, R., and Gabrys, B. (2011). Review of adaptation mechanisms for data-driven soft sensors. *Computers & chemical engineering*, 35(1):1–24.

[Kaufman et al., 2012] Kaufman, H., Barkana, I., and Sobel, K. (2012). *Direct adaptive control algorithms: theory and applications*. Springer Science & Business Media, New York.

[Kay, 2013] Kay, S. M. (2013). *Fundamentals of statistical signal processing: Practical algorithm development*, volume 3. Prentice Hall, New York.

[Khina et al., 2012] Khina, A., Ordentlich, O., Erez, U., Kochman, Y., and Wornell, G. W. (2012). Decode-and-forward for the Gaussian relay channel via standard awgn coding and decoding. In *Information Theory Workshop (ITW), 2012 IEEE*, pages 457–461. IEEE.

[Kocijan et al., 2004] Kocijan, J., Murray-Smith, R., Rasmussen, C. E., and Girard, A. (2004). Gaussian process model based predictive control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2214–2219. IEEE.

[Lacerda et al., 2016] Lacerda, M. J., Tognetti, E. S., Oliveira, R. C., and Peres, P. L. (2016). A new approach to handle additive and multiplicative uncertainties in the measurement for LPV filtering. *International Journal of Systems Science*, 47(5):1042–1053.

[Laurentys et al., 2010] Laurentys, C., Ronacher, G., Palhares, R. M., and Caminhas, W. M. (2010). Design of an artificial immune system for fault detection: a negative selection approach. *Expert Systems with Applications*, 37(7):5507–5513.

[Liu et al., 2018] Liu, X., Zhang, M., Wang, Y., and Rogers, E. (2018). Design and experimental validation of an adaptive sliding mode observer-based fault-tolerant control for underwater vehicles. *IEEE Transactions on Control Systems Technology*, pages 1–8.

[Lofberg, 2004] Lofberg, J. (2004). YALMIP: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE.

[Lopes dos Santos et al., 2011] Lopes dos Santos, P., Perdicoúlis, T. P. A., Novara, C., Ramos, J. A., and Rivera, D. E. (2011). *Linear parameter-varying system identification: New developments and trends*, volume 14. World Scientific.

[Lu and Wu, 2009] Lu, B. and Wu, F. (2009). Switching-based fault-tolerant control for an F-16 aircraft with thrust vectoring. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 8494–8499. IEEE.

[Madhag and Zhu, a] Madhag, A. and Zhu, G. Mixed ICC/H_∞ control for systems with sensors aging. *Int. J. of Control*, submitted.

[Madhag and Zhu, b] Madhag, A. and Zhu, G. Online covariance estimation of slow-varying measurement noises using a modified adaptive filter. *Int. J. Automation and Control*, *in press*.

[Madhag and Zhu, c] Madhag, A. and Zhu, G. Online sensor performance monitoring and fault detection for discrete linear parameter varying systems. *Int. J. Automation and Control*, *in press*.

[Madhag and Zhu, 2017] Madhag, A. and Zhu, G. G. (2017). Online sensor noise covariance identification using a modified adaptive filter. In *ASME 2017 Dynamic Systems and Control Conference*, pages V002T04A001–V002T04A001. American Society of Mechanical Engineers.

[Madhag and Zhu, 2018] Madhag, A. and Zhu, G. G. (2018). Guarantee performance ICC-LPV control with sensor aging. In *ASME 2018 Dynamic Systems and Control Conference*, pages V003T40A005–V003T40A005. American Society of Mechanical Engineers.

[Madhag and Zhu, 2019] Madhag, A. and Zhu, G. G. (2019). Guaranteed performance optimal control for LPV systems with aging sensors. In *IEEE2019 American Control Conference (ACC)*. IEEE.

[Mandal et al., 2017] Mandal, S., Santhi, B., Sridhar, S., Vinolia, K., and Swaminathan, P. (2017). Sensor fault detection in nuclear power plant using statistical methods. *Nuclear Engineering and Design*, 324(1):103–110.

[Mehranbod et al., 2005] Mehranbod, N., Soroush, M., and Panjapornpon, C. (2005). A method of sensor fault detection and identification. *Journal of Process Control*, 15(3):321–339.

[Meng et al., 2016] Meng, Y., Gao, S., Zhong, Y., Hu, G., and Subic, A. (2016). Covariance matching based adaptive unscented Kalman filter for direct filtering in INS/GNSS integration. *Acta Astronautica*, 120(1):171–181.

[Mohammadpour and Scherer, 2012] Mohammadpour, J. and Scherer, C. W. (2012). *Control of linear parameter varying systems with applications*. Springer Science & Business Media, New York.

[Narendra, 2012] Narendra, K. S. (2012). *Applications of adaptive control*. Elsevier, New York.

[Nesterov, 2013] Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, New York.

[Nguyen et al., 2017] Nguyen, N. T., Ting, E., Chaparro, D., Drew, M. C., and Swei, S. S.-M. (2017). Multi-objective flight control for drag minimization and load alleviation of high-aspect ratio flexible wing aircraft. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1589.

[Niksefat and Sepehri, 2002] Niksefat, N. and Sepehri, N. (2002). A QFT fault-tolerant control for electrohydraulic positioning systems. *IEEE transactions on control systems technology*, 10(4):626–632.

[Oliveira and Peres, 2009] Oliveira, R. C. and Peres, P. L. (2009). Time-varying discrete-time linear systems with bounded rates of variation: Stability analysis and control design. *Automatica*, 45(11):2620–2626.

[Ram and Davim, 2018] Ram, M. and Davim, J. P. (2018). *Diagnostic Techniques in Industrial Engineering*. s Springer International Publishing, Switzerland.

[Raol and Gopal, 2012] Raol, J. R. and Gopal, A. K. (2012). *Mobile intelligent autonomous systems*. CRC Press, New York.

[Ren and Zhu, 2011] Ren, Z. and Zhu, G. G. (2011). Modeling and control of an electric variable valve timing system for SI and HCCI combustion mode transition. In *American Control Conference (ACC), 2011*, pages 979–984. IEEE.

[Riva et al., 2018] Riva, M. H., Wielitzka, M., and Ortmaier, T. (2018). Sensitivity-based adaptive srukf for state, parameter, and covariance estimation on mechatronic systems. In *Kalman Filters-Theory for Advanced Applications*. InTech, London.

[Rodrigues et al., 2018] Rodrigues, L., Oliveira, R., and Camino, J. (2018). Parameterized LMIs for robust H_2 and H_∞ state feedback control of continuous-time polytopic systems. *International Journal of Robust and Nonlinear Control*, 28(3):940–952.

[Roman et al., 2003] Roman, T., Enescu, M., and Koivunen, V. (2003). Time-domain method for tracking dispersive channels in MIMO OFDM systems. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages 600–609. IEEE.

[Rotondo et al., 2014] Rotondo, D., Nejjari, F., and Puig, V. (2014). A virtual actuator and sensor approach for fault tolerant control of LPV systems. *Journal of Process Control*, 24(3):203–222.

[Safaei and Mahyuddin, 2017] Safaei, A. and Mahyuddin, M. N. (2017). An optimal adaptive model-free control with a Kalman-filter-based observer for a generic nonlinear MIMO system. In *Automatic Control and Intelligent Systems (I2CACIS), 2017 IEEE 2nd International Conference on*, pages 56–61. IEEE.

[Samara et al., 2008] Samara, P. A., Fouskitakis, G. N., Sakellariou, J. S., and Fassois, S. D. (2008). A statistical method for the detection of sensor abrupt faults in aircraft control systems. *IEEE Transactions on Control Systems Technology*, 16(4):789–798.

[Sharma et al., 2010] Sharma, A. B., Golubchik, L., and Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6(3):23.

[Shynk, 2012] Shynk, J. J. (2012). *Probability, random variables, and random processes: theory and signal processing applications*. John Wiley & Sons.

[Simon, 2006] Simon, D. (2006). *Optimal state estimation: Kalman, H-infinity, and nonlinear approaches*. John Wiley & Sons, New York.

[Skelton et al., 1998] Skelton, R., Iwasak, T., and Grigoriadis, K. (1998). *A unified algebraic approach to linear control design*. Taylor and Francis London.

[Skogestad and Postlethwaite, 2007] Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design*, volume 2. Wiley New York.

[Sturm, 1999] Sturm, J. F. (1999). Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653.

[Tan et al., 2018] Tan, Q., Divekar, P., Tan, Y., Chen, X., and Zheng, M. (2018). Model guided extremum seeking for diesel engine fuel injection optimization. *IEEE/ASME Transactions on Mechatronics*, PP(99):1–1.

[Tao et al., 2017] Tao, H., Paszke, W., Rogers, E., Yang, H., and Gałkowski, K. (2017). Iterative learning fault-tolerant control for differential time-delay batch processes in finite frequency domains. *Journal of Process control*, 56(1):112–128.

[Troitzsch et al., 2016] Troitzsch, C., Marten, A.-K., and Westermann, D. (2016). Non-telecommunication based dc line fault detection methodology for meshed HVDC grids. *IET Generation, Transmission & Distribution*, 10(16):4231–4239.

[Van Hessem et al., 2001] Van Hessem, D., Scherer, C., and Bosgra, O. (2001). LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 5, pages 4228–4233. IEEE.

[Wang and Xiao, 2004] Wang, S. and Xiao, F. (2004). Detection and diagnosis of AHU sensor faults using principal component analysis method. *Energy Conversion and Management*, 45(17):2667–2686.

[Wei et al., 2010] Wei, X., Verhaegen, M., and van Engelen, T. (2010). Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques. *International Journal of Adaptive Control and Signal Processing*, 24(8):687–707.

[Weingarten et al., 2006] Weingarten, H., Steinberg, Y., and Shamai, S. S. (2006). The capacity region of the Gaussian multiple-input multiple-output broadcast channel. *IEEE transactions on information theory*, 52(9):3936–3964.

[White et al., 2013a] White, A., Ren, Z., Zhu, G., and Choi, J. (2013a). Mixed H_2/H_∞ observer-based LPV control of a hydraulic engine cam phasing actuator. *IEEE Transactions on Control Systems Technology*, 21(1):229–238.

[White et al., 2012] White, A., Zhu, G., and Choi, J. (2012). A linear matrix inequality solution to the output covariance constraint control problem. In *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*, pages 163–169. American Society of Mechanical Engineers.

[White et al., 2013b] White, A., Zhu, G., and Choi, J. (2013b). A linear matrix inequality solution to the input covariance constraint control problem. In *ASME 2013 Dynamic Systems and Control Conference*, pages V002T31A002–V002T31A002. American Society of Mechanical Engineers.

[White et al., 2016] White, A., Zhu, G., and Choi, J. (2016). Optimal LPV control with hard constraints. *International Journal of Control, Automation and Systems*, 14(1):148–162.

[White et al., 2013c] White, A. P., Zhu, G., and Choi, J. (2013c). *Linear parameter-varying control for engineering applications*. Springer, New York.

[Wu and Saif, 2007] Wu, Q. and Saif, M. (2007). Robust fault detection and diagnosis for a multiple satellite formation flying system using second order sliding mode and wavelet networks. In *American Control Conference, 2007. ACC'07*, pages 426–431. IEEE.

[Xia et al., 2017] Xia, B., Shang, Y., Nguyen, T., and Mi, C. (2017). A correlation based fault detection method for short circuits in battery packs. *Journal of Power Sources*, 337(1):1–10.

[Zhai et al., 2018] Zhai, G., Wu, C., and Wang, Y. (2018). Target tracking based on millimeter wave radar in complex scenes. *International Journal of Performability Engineering*, 14(2):232.

[Zhang et al., 2017] Zhang, D., Wang, Q.-G., and Yu, L. (2017). *Filtering and Control of Wireless Networked Systems*. s Springer International Publishing, Switzerland.

[Zhang and Yan, 2001] Zhang, H. Q. and Yan, Y. (2001). A wavelet-based approach to abrupt fault detection and diagnosis of sensors. *IEEE Transactions on Instrumentation and Measurement*, 50(5):1389–1396.

[Zhang, 2002] Zhang, Q. (2002). Adaptive observer for multiple-input-multiple-output (MIMO) linear time-varying systems. *Automatic Control, IEEE Transactions on*, 47(3):525–529.

[Zhang et al., 2018] Zhang, Q., Yang, Y., Xiang, Q., He, Q., Zhou, Z., and Yao, Y. (2018). Noise adaptive Kalman filter for joint polarization tracking and channel equalization using cascaded covariance matching. *IEEE Photonics Journal*, 10(1):1–11.

[Zhang et al., 2015] Zhang, S., Yang, J. J., and Zhu, G. G. (2015). LPV modeling and mixed constrained H_2/H_∞ control of an electronic throttle. *IEEE/ASME Transactions on Mechatronics*, 20(5):2120–2132.

[Zhang et al., 2002] Zhang, X., Polycarpou, M. M., and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE transactions on automatic control*, 47(4):576–593.

[Zheng et al., 2018] Zheng, B., Fu, P., Li, B., and Yuan, X. (2018). A robust adaptive unscented Kalman filter for nonlinear estimation with uncertain noise covariance. *Sensors*, 18(3):808.

[Zhu et al., 1995] Zhu, G., Grigoriadis, K. M., and Skelton, R. E. (1995). Covariance control design for hubble space telescope. *Journal of Guidance, Control, and Dynamics*, 18(2):230–236.

[Zhu et al., 1997] Zhu, G., Rotea, M., and Skelton, R. (1997). A convergent algorithm for the output covariance constraint control problem. *SIAM Journal on Control and Optimization*, 35(1):341–361.

[Zhu et al., 1993] Zhu, G., Rotea, M. A., and Skelton, R. E. (1993). A convergent feasible algorithm for the output covariance constraint problem. In *American Control Conference, 1993*, pages 1675–1679. IEEE.

[Zhu and Skelton, 1995] Zhu, G. G. and Skelton, R. E. (1995). ℓ_2 to ℓ_∞ gains for sampled-data systems. *International Journal of Control*, 61(1):19–32.