1999

Probabilistic and Deterministic Algorithms for Information and Dynamic Systems.

Xinjia Chen
Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_disstheses/7073

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0800

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
PROBABILISTIC AND DETERMINISTIC
ALGORITHMS FOR INFORMATION AND DYNAMIC
SYSTEMS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Electrical and Computer Engineering

by
Xinjia Chen
B.S.E.E., Beijing University of Aeronautics and Astronautics, China, 1991
M.S.E.E., L.S.U., 1997
December, 1999
In Memory of My Father
ACKNOWLEDGMENTS

I would like to thank foremost my family for their support and love. I would also like to express my sincere gratitude to my Dissertation advisor and an excellent teacher, Dr. Kemin Zhou for his assistance in the process of my Dissertation research as well as many other aspects. Finally, I would like to thank Dr. J. L. Aravena, Dr. Guoxiang Gu, Dr. G. S. Ferreyra and Dr. C. Wilmot for serving on my advisory committee, and also for the knowledge they have imparted to me through various courses, without which this work would not have been possible.
# Table of Contents

**Acknowledgments** ................................................................. iii

**List of Figures** ........................................................................ vi

**Abstract** ................................................................................... viii

**Chapter**
1 **Introduction** ......................................................................... 1  
  1.1 Historical Review ............................................................... 1 
  1.2 Dissertation Outline ......................................................... 4

2 **Parallel Algorithm for Maximal Structured Singular Value** .......... 5  
  2.1 Conventional Frequency Sweeping ......................................... 6 
  2.2 Parallel Frequency Sweeping Algorithm ................................. 10 
  2.3 An Illustrative Example ....................................................... 15

3 **Fast Algorithms for Robust $\mathcal{D}$-Stability Margin** .............. 24  
  3.1 Preliminary ........................................................................... 28 
    3.1.1 Robust $\mathcal{D}$-stability Margin .................................... 28 
    3.1.2 Domain Splitting Algorithms ........................................ 29 
  3.2 Domain Splitting and Convergence ......................................... 33 
    3.2.1 An Impractical Assumption ........................................... 34 
    3.2.2 Guaranteed Critical Vertices Distance Convergence .......... 35 
  3.3 Parallel Frequency Sweeping Algorithm ................................... 40 
    3.3.1 The Main Root of Inefficiency ....................................... 40 
    3.3.2 Parallel Frequency Sweeping Algorithm ......................... 41 
  3.4 An Illustrative Example ....................................................... 47

4 **Order Statistics and Probabilistic Robust Control** .................... 56  
  4.1 Problem Formulation and Motivations .................................... 57 
    4.1.1 Robust Analysis and Optimal Synthesis ......................... 58 
    4.1.2 Quantity Range ............................................................ 60 
  4.2 Distribution Inequality ......................................................... 60 
  4.3 Performance and Confidence Tradeoff ..................................... 66 
  4.4 Minimum Sample Size ....................................................... 69 
  4.5 Quantity Range and Distribution-Free Tolerance Intervals .......... 70

5 **Constrained Optimal Synthesis and Robustness Analysis** ............ 72  
  5.1 Preliminary and Problem Formulation .................................... 74
LIST OF FIGURES

2.1 Uncertain System ................................................................. 6
2.2 A Picture of Parallel Frequency Sweeping Algorithm .................. 12
2.3 $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$ ........ 17
2.4 $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$ ........ 17
2.5 $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$ ........ 18
2.6 $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$ ........ 18
2.7 $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$ ........ 19
2.8 Domains Evaluated in Algorithm 2 ...................................... 20
2.9 Domains Evaluated in the Conventional Algorithm .................. 20
2.10 Evolution of the Global Lower Bound .................................. 22
2.11 Backward Pruning ............................................................... 22
2.12 Forward Pruning ................................................................. 23
2.13 Present Pruning ................................................................. 23
3.1 Robust $\mathcal{D}$-Stability ..................................................... 26
3.2 The Image of the Last Subdomain ....................................... 40
3.3 A Picture of Parallel Frequency Sweeping Algorithm .................. 43
3.4 $k_m$ Upper Bound (solid) and Lower Bound (dashed) on $\mathcal{Q}$ ........ 50
3.5 $k_m$ Upper Bound (solid) and Lower Bound (dashed) on $\mathcal{Q}$ ........ 50
3.6 $k_m$ Upper Bound (solid) and Lower Bound (dashed) on $\mathcal{Q}$ ........ 51
3.7 $k_m$ Upper Bound (solid) and Lower Bound (dashed) on $\mathcal{Q}$ ........ 51

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8</td>
<td>Domains Evaluated in Algorithm 2</td>
<td>52</td>
</tr>
<tr>
<td>3.9</td>
<td>Domains Evaluated in the Conventional Algorithm</td>
<td>52</td>
</tr>
<tr>
<td>3.10</td>
<td>Evolution of the Global Upper Bound</td>
<td>53</td>
</tr>
<tr>
<td>3.11</td>
<td>Evolution of the Global Upper Bound</td>
<td>53</td>
</tr>
<tr>
<td>3.12</td>
<td>Backward Pruning</td>
<td>54</td>
</tr>
<tr>
<td>3.13</td>
<td>Forward Pruning</td>
<td>54</td>
</tr>
<tr>
<td>3.14</td>
<td>Present Pruning</td>
<td>55</td>
</tr>
<tr>
<td>4.1</td>
<td>Uncertain System</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Synthesis Framework</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>Confidence Level</td>
<td>68</td>
</tr>
<tr>
<td>5.1</td>
<td>A Picture of $F_u$</td>
<td>86</td>
</tr>
</tbody>
</table>
ABSTRACT

The central goal of control engineering is to assure the robust stability and performance for systems in the presence of uncertainties. This thesis deals with these fundamental issues in three different frameworks. First, the robustness of uncertain systems are discussed in the structured singular value \( \mu \) framework. Parallel algorithms are developed which greatly facilitates robustness analysis. Second, the robust control problems are tackled in the Kharitonov framework. Efficient Algorithms have been developed for computing the robust \( \mathcal{D} \)-stability margin for arbitrary root domain \( \mathcal{D} \). This allows for a more sophisticated analysis of system robustness. Finally, aimed at breaking through the barrier of NP hardness and reducing conservativeness, the robust control problems are considered in the probabilistic framework. Minimum computational effort for robust analysis with a certain degree of reliability is investigated and related sample sizes are derived. An interesting link between classic order statistics theory and robust control is established. Moreover, the classic order statistics distribution theory is generalized to accommodate discontinuous populations.
CHAPTER 1

INTRODUCTION

In many disciplines of engineering it is often convenient, for the analysis and design purposes, to approximate the real behavior of physical systems by mathematical models. For some applications, however, and in particular when one wishes to design a high performance controller, the difference between the behavior of the mathematical model and the physical system can be crucial to the performance of the final design. The theory of robust control attempts to take into account these inherent inaccuracies in the model, and provides systematic analysis and design techniques in the face of this "uncertainty".

1.1 HISTORICAL REVIEW

Historically, robust control theory is formulated on the worst case deterministic paradigm. Based on this paradigm, two different frameworks have been developed. One is the structured singular value $\mu$ framework. The advantage of $\mu$ framework is its generality in dealing with arbitrary system structure and uncertainty structure. The central issue is the computation of $\mu$. This is because the analysis of robust stability and performance with structured uncertainty boils down to the problem of computing the supremum of the structured singular value over all frequencies $[21, 53]$. That is, $\mu_{\text{max}} := \sup_{\omega \in \mathbb{R}} \mu_{\Delta}(M(j\omega))$ where $M(s)$ is the transfer function of the generalized system and $\Delta$ is a set of block structured uncertainties. Existing techniques aimed at tightly bounding $\mu$ for each frequency and thus most of the computational effort is wasted. In this research work, we have developed a parallel
frequency sweeping algorithm which computes the maximal structured singular value \( \mu \) without tightly bounding \( \mu \) for each frequency and thus significantly reduce the computational complexity [18].

Another framework is the Kharitonov framework. The advantage of this framework is that a more general stability concept, namely, \( D \)-stability, can be discussed. This allows for a more sophisticated analysis and design for uncertain systems. The central issue of this framework is the stability margin \( k_m \). The drawback of conventional techniques is its efficiency associated with frequency sweeping and domain splitting. In this research, we develop a smart frequency sweeping strategy and domain splitting techniques which significantly improve the efficiency [19].

In recent years, research works in computational complexity show that many deterministic worst-case robust analysis and synthesis problems are NP hard, which means that the exact analysis and synthesis of the corresponding robust control problems may be computational demanding [9, 46, 53]. Moreover, the deterministic worst-case robustness measures may be quite conservative due to overbounding of the system uncertainties. As pointed out by Khargonekar and Tikku in [33], the difficulties of deterministic worst-case robust control problems are inherent to the problem formulations and a major change of the paradigm is necessary. An alternative to the deterministic approach is the probabilistic approach which has been studied extensively by Stengel and co-workers, see for example, [39], [43], [35] and references therein. Aimed at breaking through the NP-hardness barrier and reducing the conservativeness of the deterministic robustness measures, the probabilistic approach has recently received a renewed attention in the work by Barmish and Lagoa [8], Barmish, Lagoa, and Tempo [6], Barmish and Polyak [7], Khargonekar and Tikku [33], Bai, Tempo, and Fu [5], Tempo, Bai, and Dabbene [45], Yoon and
Khargonekar [50], Zhu, Huang and Doyle [51], Chen and Zhou [13, 14, 15, 16, 17] and references therein.

In particular, Tempo, Bai and Dabbene [45] and Khargonekar and Tikku [33] have derived bounds for the number of samples required to estimate the upper bound of a quantity with a certain a priori specified accuracy and confidence. It is further shown that this probabilistic approach for robust control analysis and synthesis has low complexity [33, 45]. It should also be pointed out that the uncertain parameters do not necessarily have to be random, they can be regarded as randomized variables as pointed out by Khargonekar and Tikku [33].

In general, an exact probabilistic robustness analysis or design is very hard if not impossible. Therefore, the probabilistic approach has been developed in two different directions. One direction focuses on obtaining hard bounds so that certain property of the uncertain system holds with a given probability, see a discussion of this topic in [51]. For example, Barmish and Polyak [7] proposed the concept of value set predictor, though no probabilistic bound is derived; in another paper [8], Barmish and Lagoa have given results on worst-case probability distributions for certain types of robustness analysis problems. In general, obtaining a good hard bound requires more detailed mathematical structure of the problems. For example, Barmish and Lagoa in [8] had made symmetric and nonincreasing assumptions on uncertain parameters. Another direction focuses on obtaining soft bounds [51], which means that with a certain confidence level, a property of the uncertain system holds with a certain probability. The way to get the soft bound is by doing Monte Carlo simulation. The beauty of this approach is that it can be applied to any system with complex structure and there is little growth in the computational cost with the number of parameters.
In this research work, the robust control problems are also considered in the probabilistic framework. Minimum computational effort for robust analysis with a certain degree of reliability is investigated and related sample sizes are derived. An interesting link between classic order statistics theory and robust control is established. Moreover, the classic order statistics distribution theory is generalized to accommodate discontinuous populations.

1.2 Dissertation Outline

The dissertation is organized as follows. Chapter 2 presents the parallel algorithms for robust control in the $\mu$ framework. Chapter 3 considers the robust control problems in the Kharitonov framework. Efficient techniques for computing robust $\mathcal{D}$-stability margin are developed. Chapter 4 and 5 study robustness issues in the probabilistic framework. Chapter 4 focuses on unconstrained problems while Chapter 5 study constrained problems. Chapter 6 gives some concluding remarks.
In this chapter, we have developed a parallel frequency sweeping algorithm which computes the maximal structured singular value $\mu$ without tightly bounding $\mu$ for each frequency and thus significantly reduce the computational complexity.

It is well known that the analysis of robust stability and performance with structured uncertainty boils down to the problem of computing the supremum of the structured singular value over all frequencies [21, 53]. That is, $\mu_{\text{max}} := \sup_{\omega \in \mathbb{R}} \mu_{\Delta}(M(j\omega))$ where $M(s)$ is the transfer function of the generalized system and $\Delta$ is a set of block structured uncertainties. Related to this problem are the interesting approaches of Helmersson [28] and Lawrence, Tits and Dooren [34], where upper bounds for $\mu$ over a frequency interval are computed (see also the approximate, but computationally cheaper algorithm in Feron [25]). However, for the precise computation of the maximal structured singular value $\mu_{\text{max}}$, the existing techniques fall into an unique format, that is, choose and grid a range of frequency, then compute $\mu$ for each frequency and find the maximum [2]. Since the exact computation is in general impossible, $\mu$ is obtained for each frequency by tightly bounding. Sophisticated upper bounds and lower bounds have been derived for example in [4, 21, 23, 37] and techniques such as branch and bound [36] have been developed to refine the bounds.

It is noted that the existing techniques for computing the maximal structured singular value $\mu_{\text{max}}$ lack of efficiency because of the tedious frequency sweeping. In this chapter, we investigate a smart frequency sweeping strategy. More specifically,
we apply branch and bound scheme to compute $\mu$ for $N > 1$ frequencies in parallel. We introduce a powerful "pruning" mechanism, which eliminates any branch with upper bound smaller than $\frac{\hat{\mu}}{1-\epsilon}$ where $\hat{\mu}$ is the maximum record of the lower bounds of all branches ever generated and $\epsilon > 0$ is the tolerance. The final $\hat{\mu}$ is returned as the maximal structured singular value $\mu_{\text{max}}$. Since $\hat{\mu}$ is the maximum record of the lower bounds obtained in all branches generated (no matter belong to the same frequency or not), it will increase much faster than its counterpart in the conventional frequency sweeping algorithms. Note that the raise of $\hat{\mu}$ results in a significant number of branches to be pruned. Thus $\hat{\mu}$ convergences quickly to the maximal structured singular value $\mu_{\text{max}}$.

This chapter is organized as follows. Section 2.1 discusses existing techniques for computing the maximal structured singular value. Section 2.2 presents our Parallel Frequency Sweeping Algorithm. An illustrative example is provided in Section 2.3.

2.1 CONVENTIONAL FREQUENCY SWEEPING

Most robustness analysis problems can be put in an $M - \Delta$ setup as shown in Figure 2.1. Let $M(s)$ be a stable transfer matrix. Let $\Delta$ be a set of block structured uncertainties as follows:

$$\Delta := \text{diag}\{\Delta_1, \ldots, \Delta_f, \delta_1 I, \ldots, \delta_m I, r_1 I, \ldots, r_k I\}$$

![Figure 2.1: Uncertain System](image-url)
where
\[ \Delta_i \in \mathbb{C}^{n \times n}, \quad \delta_j \in \mathbb{C}, \quad \tau_\ell \in \mathbb{R} \]

The structured singular value \( \mu \) is defined as:
\[
\mu_{\Delta}(M) := \left( \min\{\sigma(\Delta) : \det(I - M\Delta) = 0\} \right)^{-1}.
\]

We consider the computation of
\[
\mu_{\text{max}} := \sup_{\omega \in \mathbb{R}} \mu_{\Delta}(M(\omega)).
\]

For the simplicity of notation, let \( M(\omega) = M(j\omega) \). Then \( \mu_{\text{max}} = \sup_{\omega \in \mathbb{R}} \mu_{\Delta}(M(\omega)) \).

In practice, it is impossible to search \( \mu_{\text{max}} \) over all frequencies. However, we can estimate \( \mu_{\text{max}} \) as follows.

Choose a range of frequency \([\omega_l, \omega_u] \in \mathbb{R}\) and grid it as
\[
\omega_j = \omega_l + \left( \omega_u - \omega_l \right) \frac{j - 1}{NK}, \quad j = 1, \ldots, NK
\]
where \( N \geq 2 \) and \( K \geq 1 \) are integers (In practice, gridding is usually based on the logarithmic scale. However, in this chapter, we use uniform gridding for the simplicity of description). Then an estimate for \( \mu_{\text{max}} \) can be defined as
\[
\tilde{\mu}_{\text{max}} := \max_{j=1,\ldots,NK} \mu_{\Delta}(M(\omega_j)).
\]

Define the (maximum positive real eigenvalue) function \( \bar{\lambda}_R : \mathbb{C}^{n \times n} \to \mathbb{R} \) as
\[
\bar{\lambda}_R(M) := \max\{\lambda : \lambda \text{ is a positive real eigenvalue of } M\}.
\]
with $\tilde{\lambda}_R(M) = 0$ if $M$ has no positive real eigenvalues. Let $\mathbf{B} \Delta := \{ \Delta \in \Delta : \bar{\sigma}(\Delta) \leq 1 \}$. Then

$$\mu_\Delta(M) = \max_{\Delta \in \mathbf{B} \Delta} \tilde{\lambda}_R(M \Delta).$$

Let $Q \subset \mathbf{B} \Delta$. Define $\mu$ on a box [36]

$$\mu(M, Q) := \max_{\Delta \in Q} \tilde{\lambda}_R(M \Delta).$$

There exist techniques in [36] for computing an upper bound $UB(M, Q)$ and a lower bound $LB(M, Q)$ for $\mu(M, Q)$. Thus a branch and bound scheme can be applied to compute $\mu_\Delta(M)$ with parameter space $\mathbf{B} \Delta$.

To the best of our knowledge, no effort in the existing literature has been devoted to exploit a smart frequency sweeping strategy. Existing techniques work essentially as follows.

For $j = 1, \cdots, NK$, apply the following Algorithm 1 to compute an upper bound $UB^j$ and a lower bound $LB^j$ for $\mu_\Delta(M(\omega_j))$ such that $UB^j - LB^j \leq \varepsilon$. Then $\tilde{\mu}_{\text{max}}$ satisfies

$$\max_{j=1,\ldots,NK} LB^j \leq \tilde{\mu}_{\text{max}} \leq \max_{j=1,\ldots,NK} UB^j$$

where

$$\max_{j=1,\ldots,NK} UB^j - \max_{j=1,\ldots,NK} LB^j \leq \varepsilon.$$

**Algorithm 1 — Branch and Bound** ([36])

*Initialize* Let $U_j = \{Q_k\} = \mathbf{B} \Delta$.

*Let*

$$UB^j = \max_k UB(M(\omega_j), Q_k),$$
\[ \text{while } UB^j - LB^j > \varepsilon \]

- Choose \( Q \) to be any element of \( U_j \) with \( UB(M(\omega_j), Q) = UB^j \).
- Partition \( Q \) into \( Q_a \) and \( Q_b \) by bisecting along one of its longest edges.
- Add \( Q_a \) and \( Q_b \) into \( U_j \). Remove \( Q \) from \( U_j \).
- Remove from \( U_j \) any \( Q \) with

\[ UB(M(\omega_j), Q) < LB^j. \] (2.3)

\textit{endwhile}

The most important mechanism of Algorithm 1 is “pruning” [36]. That is, any element of \( U_j \) for which (2.3) is satisfied will never again be partitioned and need not be considered further. We call inequality (2.3) as the “pruning condition”.

We can see that existing techniques for computing \( \bar{\mu}_{\text{max}} \) employ branch and bound techniques for each frequency independently. In particular, the pruning process for one frequency is independent of another. \( \mu_\Delta(M) \) is bounded tightly for each frequency. Note that we usually need to evaluate \( \mu_\Delta(M) \) for many frequencies in order to obtain a reasonably good estimate of the maximal structured singular value \( \mu_{\text{max}} \). Thus the overall computation is still a heavy burden, even though the computation of \( \mu_\Delta(M) \) for each frequency is very efficient.

Thus for the sake of efficiency, there is a strong motivation to develop a smart frequency sweeping strategy. More specifically, we would raise the following question:
Is it possible to obtain the maximal structured singular value $\mu_{\text{max}}$ without tightly bounding $\mu_\Delta(M(\omega_j))$ for each frequency $\omega_j$?

The following section is devoted to answering this question.

2.2 Parallel Frequency Sweeping Algorithm

It is fair to compare the performance of different algorithms on the same set of frequencies. Therefore, we consider again frequencies $\omega_j, \ j = 1, \cdots, NK$ defined by (2.1) and relabel them as

$$
\omega_{ij} := \omega_i + \frac{(\omega_u - \omega_i)(i - 1)}{N} + \frac{(\omega_u - \omega_i)(j - 1)}{NK}, \quad i = 1, \cdots, N, \ j = 1, \cdots, K.
$$

Now we are in a good position to present our Parallel Frequency Sweeping Algorithm as follows.

**Algorithm 2 — Parallel Frequency Sweeping Algorithm**

- **Step 1:** Initialize. Set $j = 1$. Set $\hat{\mu} = 0$. Set tolerance $\epsilon > 0$. Set maximal iteration number $IT$.

- **Step 2:** Update $\hat{\mu}$ and record the number of iterations $r(j)$ for frequency $\omega_{ij}$ by the following steps.

  - Step 2-1: Let $\mathcal{U}_{ij} = \{Q_k\} = \mathbf{B}_\Delta, \ i = 1, \cdots, N$. Set $r = 1$.

  - Step 2-2: If $r = IT + 1$ or $\mathcal{U}_{ij}$ is empty for any $i \in \{1, \cdots, N\}$ then record $r(j) = r$ and go to Step 3, else do the following for all $i$ such that $\mathcal{U}_{ij}$ is not empty.

    * Choose $Q$ to be any element of $\mathcal{U}_{ij}$ with

      $$
      UB(M(\omega_{ij}), Q) = \max_{Q_k \in \mathcal{U}_{ij}} UB(M(\omega_{ij}), Q_k).
      $$

      * Choose $Q$ to be any element of $\mathcal{U}_{ij}$ with
* Partition $Q$ into $Q_a$ and $Q_b$ by bisecting along one of its longest edges.

* Add $Q_a$ and $Q_b$ into $U_{ij}$. Remove $Q$ from $U_{ij}$.

* Update

$$\hat{\mu} = \max\{\hat{\mu}, \ LB(M(\omega_{ij}), Q_a), \ LB(M(\omega_{ij}), Q_b)\}. \quad (2.4)$$

* Remove from $U_{ij}$ any $Q$ with

$$UB(M(\omega_{ij}), Q) < \frac{\hat{\mu}}{1 - \epsilon}. \quad (2.5)$$

- Step 2–3: Set $r = r + 1$ and go to Step 2–2.

• Step 3: If $j = K$ then STOP, else set $j = j + 1$ and go to Step 2.

In Algorithm 2, $N$ branches of frequency sweeping are performed in parallel with starting frequencies $\omega_{i1}, \ i = 1, \cdots, N$ and step size $\frac{\omega_n - \omega_1}{NK}$. Also, a branch and bound scheme is applied to compute $\mu$ for $N$ frequencies in parallel. Any branch with upper bound smaller than $\frac{\hat{\mu}}{1 - \epsilon}$ will be pruned, where $\hat{\mu}$ is the maximum record of the lower bounds of all branches ever generated. The final $\hat{\mu}$ is returned as the maximal structured singular value $\mu_{\text{max}}$. Algorithm 2 is visualized in the following Figure 2.2 for the case of $N = 3, \ K = 4$.

Remark 2.1 Note that Algorithm 2 provides a substantial improvement on efficiency over existing algorithms in computing the maximal structured singular value. This can be explained by the significant relaxation in the "pruning condition" of Algorithm 2. To see the difference of the two "pruning conditions", we can compare the right hand sides of inequalities (2.5) and (2.3). By (2.4) and (2.2), we can see
that $\frac{\hat{\mu}}{1-\varepsilon}$ can be much larger than $LB^j$. This is because $\hat{\mu}$ is the maximum record of the lower bounds obtained in all branches of all frequencies evaluated and being evaluated, while $LB^j$ is only the maximum record of the lower bounds obtained in branches of the frequency being evaluated. Moreover, $\hat{\mu}$ is enlarged to $\frac{\hat{\mu}}{1-\varepsilon}$ in the "pruning condition" (2.5) and hence the "pruning" process is further facilitated. The significant relaxation of the 'pruning condition' leads to a substantial decrease of the number of total subdomains needed to be evaluated. Therefore, our algorithm is much more efficient than those previously available to control engineers.

**Remark 2.2** It is important to note that Algorithm 2 involves only one CPU processor. It is fundamentally different from the parallel algorithms which involves more than one CPU processors.

**Remark 2.3** A substantial amount of computation can be saved by the following mechanisms. First, further computation of the lower bound on a domain is not needed once it is determined that the lower bound is smaller than the existing global
lower bound. This can be seen from equation (2.4). Second, the computation of the upper bound should be terminated once condition (2.5) is satisfied. The idea of these two mechanisms is to avoid as much as possible the tightly computation of the lower bound and the upper bound.

In addition to the novel frequency sweeping strategy, another character of Algorithm 2 is that there is no tolerance criteria directly enforced on the final result, however, the final result falls into tolerance automatically.

**Theorem 2.1** Suppose that the maximal iteration number $I_T < \infty$ and that Algorithm 2 stops with $r(j) \leq I_T$, $j = 1, \cdots, K$. Then the final $\tilde{\mu}$ satisfies

$$0 \leq \frac{\hat{\mu}_{\text{max}} - \tilde{\mu}}{\hat{\mu}_{\text{max}}} < \epsilon.$$ 

**Proof.** Since $\hat{\mu}$ is the maximal record of the lower bounds, we have $\hat{\mu}_{\text{max}} - \tilde{\mu} \geq 0$. We only need to show that $\frac{\hat{\mu}_{\text{max}} - \tilde{\mu}}{\hat{\mu}_{\text{max}}} < \epsilon$. By the assumption that Algorithm 2 stops with $r(j) \leq I_T$, $j = 1, \cdots, K$, we know that all subdomains ever generated are finally removed because the “pruning condition” (2.5) is satisfied. Note that there exists a subdomain $Q_{ij}$ for frequency $\omega_{ij}$ such that $\mu(M(\omega_{ij}), Q_{ij}) = \hat{\mu}_{\text{max}}$. Let $\tilde{\mu} = \hat{\mu}$ when $Q_{ij}$ is removed. Then $\tilde{\mu}_{\text{max}} \leq UB(M(\omega_{ij}), Q_{ij}) < \frac{\hat{\mu}}{1 - \epsilon}$. Note that $\tilde{\mu}$ is nondecreasing thus the final $\tilde{\mu} \geq \hat{\mu}$. It follows that

$$\tilde{\mu}_{\text{max}} < \frac{\hat{\mu}}{1 - \epsilon} \implies \frac{\hat{\mu}_{\text{max}} - \tilde{\mu}}{\hat{\mu}_{\text{max}}} < \epsilon.$$ 

The proof is thus completed. \qed
Note that one important concern of an algorithm is convergence. It is usually desirable that, given any tolerance \( \epsilon > 0 \), an algorithm stops and returns the result withiin tolerance in a finite number of iterations. Obviously, the convergence requirement imposes condition on the quality of bounds.

**Definition 2.1** The upper bound \( UB(M,.) \) and lower bound \( LB(M,.) \) are said to be continuous if

\[
\lim_{d(Q) \to 0} UB(M, Q) - LB(M, Q) = 0
\]

where \( d(Q) := \max_{q, q' \in Q} ||q - q'|| \) with \( Q \subseteq B \Delta \).

**Theorem 2.2** Suppose that all the upper bounds and lower bounds are continuous and that there is at least one nonzero lower bound appears after a finite number of iterations. Let the maximal iteration number \( IT = \infty \). Then, for arbitrary tolerance \( \epsilon > 0 \), Algorithm 2 stops with a finite number of domain partitions for each \( j \), i.e., \( r(j) < \infty, \ j = 1, \cdots, K \). Moreover, the final \( \mu \) satisfies

\[
0 < \frac{\bar{\mu}_{max} - \mu}{\bar{\mu}_{max}} < \epsilon.
\]

**Proof.** Suppose that Algorithm 2 does not stop with a finite number of domain partitions for each \( j \). Then there exists \( \omega_{ij} \) and an infinite sequence of nested subdomains \( \{Q^i_r\} \) associated with frequency \( \omega_{ij} \) such that \( Q^i_1 \supset Q^i_2 \supset \cdots \supset Q^i_r \supset \cdots \). Note that by the assumption \( \exists r_0 < \infty, \mu_0 > 0 \) such that \( \mu \geq \mu_0, \ \forall r > r_0 \). By the continuity, \( \exists r_1 \) such that

\[
UB(M(\omega_{ij}), Q^i_r) - LB(M(\omega_{ij}), Q^i_r) < \frac{\epsilon}{1 - \epsilon} \mu_0, \ \forall r > r_1.
\]
Let $r_2 = \max\{r_0, r_1\} + 1$. Then

$$UB(M(\omega_{ij}), Q_{r_2}^{ij}) - LB(M(\omega_{ij}), Q_{r_2}^{ij}) < \frac{\epsilon}{1 - \epsilon} \mu_0.$$ 

Thus $UB(M(\omega_{ij}), Q_{r_2+1}^{ij}) - \hat{\mu} < \frac{\epsilon}{1 - \epsilon} \hat{\mu} \implies UB(M(\omega_{ij}), Q_{r_2+1}^{ij}) < \frac{\hat{\mu}}{1 - \epsilon}$ which implies that $Q_{r_2+1}^{ij}$ is removed. This is a contradiction. Therefore Algorithm 2 stops with a finite number of domain partitions for each $j$ and hence by the same argument as in the proof of Theorem 2.1, we have $0 \leq \frac{\hat{\mu}_{\max} - \hat{\mu}}{\mu_{\max}} < \epsilon$. The proof is thus completed.

$$\square$$

### 2.3 An Illustrative Example

Our computational experience shows that Algorithm 2 provides a significant improvement over the conventional algorithms for most control problems. Moreover, the improvement depends on the problems and can be arbitrarily good.

Consider an $M - \Delta$ set up as shown in Figure 2.1 where $\Delta = \text{diag}(\delta_1, \delta_2) \in \mathbb{R}^{2 \times 2}$ and $M(s) = C(sI - A)^{-1}B$ with

$$A = \begin{bmatrix} -1 & -10 & -1 & 10 \\ -0.5 & -1 & 1 & 0.5 \\ 0.5 & -4 & -1 & -10 \\ -10 & 0.5 & 0 & -2.5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.5 \end{bmatrix}.$$ 

The original domain for each frequency is $\mathcal{B} \Delta = [-1, 1] \times [-1, 1]$. The upper bound and lower bound of $\mu$ on $\mathcal{B} \Delta$ is shown in Figures 2.3, 2.4, 2.5, 2.6 and 2.7. We
can see that for most of the frequencies the upper bound and lower bound are far apart and thus the importance of branch and bound techniques is obvious.

To compute the supremum of \( \mu \), we uniformly grid frequency band \([0.01, 15.01]\) and obtain 1,500 grid frequencies as

\[ \omega_j = 0.01j, \quad j = 1, \ldots, 1500. \]

In Algorithm 2, we choose the relative error \( \varepsilon = 0.01 \) and \( N = 30, \quad K = 50 \). The 1,500 frequencies are regrouped as

\[ \omega_{ij} = 0.01 + 0.2(i - 1) + 0.01(j - 1), \quad i = 1, \ldots, 30; \quad j = 1, \ldots, 50. \]

We ran the program in Sun Spark 10 workstation. The running time is 83 seconds. The total number of domains evaluated is 1,570. We obtained \( \hat{\mu} = 0.6952 \) which is achieved at frequency \( \omega_{20,18} = 9.78 \). By Theorem 2.1, we can concluded that

\[ 0 \leq \frac{\tilde{\mu}_{\text{max}} - \hat{\mu}}{\tilde{\mu}_{\text{max}}} < \varepsilon \quad \iff \quad 0 \leq \tilde{\mu}_{\text{max}} - \hat{\mu} < \frac{\varepsilon \hat{\mu}}{1 - \varepsilon} = \frac{0.01 \times 0.6952}{1 - 0.01} \approx 0.007. \]

To compare the performance of the conventional algorithm with that of Algorithm 2, it is fair to choose the tolerance \( \varepsilon = 0.007 \) in Algorithm 1. We also ran the program in Sun Spark 10 workstation. The running time is about 6 hours. The total number of domains evaluated is 11,208. We obtained \( \hat{\mu} = 0.6954 \) which is achieved at frequency \( \omega_{9783} = 9.783 \). Therefore, Algorithm 2 has a speed-up of 260 over the conventional algorithm. Moreover, the number of domains evaluated in Algorithm 2 is only a small fraction (which is \( \frac{1570}{11208} \approx 0.13 \)) of that of the conventional algorithm.
Figure 2.3: $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$

Figure 2.4: $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$
Figure 2.5: $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$

Figure 2.6: $\mu$ Upper Bound (dashed) and Lower Bound (solid) on $B\Delta$
The number of domains evaluated in Algorithm 2 and the conventional one for each frequency is shown respectively in Figure 2.8 and Figure 2.9.

We can see that Algorithm 2 provides much superior performance than the conventional algorithms. The improvement comes from the characteristic pruning mechanism in Algorithm 2. More formally, we describe the pruning process in Algorithm 2 as follows.

Let a record of global lower bound be denoted as $(L, \omega)$ where $L$ is the value of the global lower bound achieved by frequency $\omega$. Let $Q_{ij} \subseteq \mathbf{B} \Delta$ be a domain associated with frequency $\omega_{ij}$. When $Q_{ij}$ is eliminated, i.e., $UB(M(\omega_{ij}), Q_{ij}) < \frac{L}{1-\varepsilon}$ is satisfied, there are only three cases as follows.

- Case (i): $\omega_{ij} < \omega$. We call the elimination as **Backward Pruning**.
- Case (ii): $\omega_{ij} > \omega$. We call the elimination as **Forward Pruning**.
- Case (iii): $\omega_{ij} = \omega$. We call the elimination as **Present Pruning**.
Figure 2.8: Domains Evaluated in Algorithm 2

Figure 2.9: Domains Evaluated in the Conventional Algorithm
All the above three types of pruning processes play important roles in Algorithm 2. However, there is only Present Pruning in the conventional algorithm. Therefore, Algorithm 2 has a much more powerful pruning mechanism and is much more efficient.

In this example, we have 24 records which are shown in Figure 2.10, where the vertical coordinate represents the record value and the horizontal coordinate represents the frequency achieving it. Two consequent records are connected by dashed line. The effectiveness of the three types of pruning processes are shown respectively in Figures 2.11, 2.12 and 2.13, where the vertical coordinate represents the number of domains eliminated by the record and the horizontal coordinate represents the record index.
Figure 2.10: Evolution of the Global Lower Bound

Figure 2.11: Backward Pruning
Figure 2.12: Forward Pruning

Figure 2.13: Present Pruning
CHAPTER 3

FAST ALGORITHMS FOR ROBUST \( \mathcal{D} \)-STABILITY MARGIN

In Chapter 2 we have considered the robust stability problems in the \( \mu \) framework. In this Chapter, we consider a more general concept of robust stability. In particular, we study the computation of robust \( \mathcal{D} \)-stability margin \( k_m \), which equals to \( \frac{1}{\mu} \) for the special case when the root domain \( \mathcal{D} \) is defined as the open left half complex plane.

In this Chapter we consider the robust \( \mathcal{D} \)-stability margin problem under polynomial structured real parametric uncertainty. Based on the work of de Gaston and Safonov (1988), we have developed techniques such as, a parallel frequency sweeping strategy, different domain splitting schemes, which significantly reduce the computational complexity and guarantee the convergence.

Robustness of control systems has been one of the central issues in the control community in the last two decades. Most of the research efforts have been devoted to the \( \mu \) framework\[2, 3, 21, 36, 53\] and the Kharitonov framework\[4, 31, 44\]. One of the well studied robustness analysis problem is the computation of robust stability margin under polynomial structured real parametric uncertainty. A number of different approaches have been proposed in the Kharitonov framework aiming at the nonconservative computation of the robust stability margin. Among these, we recall the geometric programming methods \[49\], the algorithm based on the Routh table \[42\], and the domain splitting approach \[26\] based on the Zero Exclusion Condition\[24\]. In general, the algorithms in \[42, 49\] is more efficient than the al-
gorithm in [26]. The main reason is that the algorithms in [42, 49] are essentially based on the Routh-Hurwitz criterion and thus only finite conditions need to be evaluated, while the algorithm in [26] is based on the Zero Exclusion Condition and thus a frequency sweeping is essential.

Even though a frequency sweeping is a necessity, an algorithm based on the Zero Exclusion Condition has its particular advantage when dealing with robust $D$-stability problems. For example [4], for high order control systems, a typical specification might be as follows: The closed loop polynomials should have a pair of "dominant roots" in disks of given radius $\epsilon > 0$ centered at $z_{1,2} = -u \pm jv$, and all remaining roots having real part less than $-\sigma$ with $\sigma > 0$ (See Figure 3.1, where $z_1 \in D_1$, $z_2 \in D_2$, $D = D_1 \cup D_2 \cup D_3$). Then, a robust $D$-stability margin problem can be defined as follows: What is the maximum perturbation of plant parameters such that the roots of the closed loop polynomial remain robustly in $D = \{z \in \mathbb{C} : |z - z_1| < \epsilon\} \cup \{z \in \mathbb{C} : |z - z_2| < \epsilon\} \cup \{z \in \mathbb{C} : \Re(z) < \sigma\}$?

Since the root region $D$ can be defined as a union of disjoint open subsets with complicated boundary in the complex plane, the robust $D$-stability problems can not be solved, in general, by existing results in the $\mu$ framework or the algorithms in [42, 49] which are based on the Routh-Hurwitz criterion. For special cases that $D$ is simply connected and is defined via the Nyquist curve of certain rational polynomials $f(s) = \frac{g(s)}{h(s)}$, the robust $D$-stability problem of $p(s)$ may be reduced to the robust stability problem of polynomial $\tilde{p}(s) = p(f(s)).(h(s))^{n_h}$ where $n_h$ is the degree of polynomial $h(s)$ and then the algorithms in [42, 49] may be applied. However, the complexity is increased substantially because the coefficients of $\tilde{p}(s)$ may be complex and the degree of $\tilde{p}(s)$ is $n_g$ times of the degree of $p(s)$ where $n_g$ is the degree of polynomial $g(s)$ [41].
The advantage of an algorithm based on the Zero Exclusion Condition is that it can be applied to the robust $\mathcal{D}$-stability problem with arbitrary complicated root region $\mathcal{D}$. What we only need to do is to verify whether the Zero Exclusion Condition is satisfied for all boundary point of $\mathcal{D}$. Note that we usually need to evaluate the Zero Exclusion Condition for many boundary points of $\mathcal{D}$ to come up with a reasonably accurate solution. Therefore, there is strong motivation to develop efficient algorithms based on the Zero Exclusion Condition to tackle the robust $\mathcal{D}$-stability problems.

The algorithm proposed by de Gaston and Safonov [26] is based on the Zero Exclusion Condition and thus can be applied to the general robust $\mathcal{D}$-stability problem. However, there are two problems with that algorithm.

First of all, it is noted that the convergence of the algorithm in [26] was concluded upon an impractical assumption. That is, a domain can be divided fine enough to converge to a point (see [26] line 40 – 53 of page 156 in the proof of the Convergence
Theorem). However, to satisfy this assumption, the computational complexity may be unacceptably high. In this chapter, we have shown that it is sufficient to guarantee the convergence in computing the stability margin by guaranteeing that the distance between critical vertices converge to 0. Therefore, it is not necessary to divide a subdomain so many times to collapse it to a point. In contrast, what we need is to make the critical vertices crunch together. Thus, the computation can be reduced greatly. We provide two splitting schemes which guarantee this.

Another problem with the algorithm in [26] is its inefficiency. One main hurdle is its tedious frequency sweeping. Consider a family of uncertain polynomials \( p(s, q) \), \( q \in Q \) where \( Q \) is the set of uncertain parameters. Let \( k_m(\omega, Q) = \sup \{ k : 0 \notin p(j\omega, kQ) \} \) where \( p(j\omega, kQ) \) is the value set associated with frequency \( \omega \) and perturbation bound \( k \). The algorithms in [26] compute \( k_m(\omega, Q) \) exactly for each frequency \( \omega \) and compare to find the minimum as the stability margin. To the best of our knowledge, all frequency sweeping techniques in the literature follow this format.

In this chapter, we investigate a smart frequency sweeping strategy. We compute \( k_m \) for \( n_r > 1 \) frequencies in parallel. Domain splitting is also performed in parallel at each iteration level. Information is exchanged among all subdomains to determine which subdomain for which frequency should be eliminated from further consideration without obtaining the exact value of \( k_m \). The stability margin is achieved as the minimum record of the upper bounds of all the subdomains ever generated. The convergence rate is much faster than that of [26].

This chapter is organized as follows. Section 3.1 introduce the robust \( \mathcal{D} \)-stability problem and the work of de Gaston and Safonov [26]. Section 3.2 discusses the Convergence Theorem of [26] and different domain splitting schemes. Section 3.3
presents our Parallel Frequency Sweeping Algorithm. An illustrative example is given in Section 3.4.

3.1 PRELIMINARY

It is well known that the stability problem of an MIMO system can be reduced to the study of the root location of a related polynomial [4, 12]. We consider a family of polynomials \( p(s, q) \) of degree \( n \) whose coefficients \( a_i(q) \) are continuous functions of \( \ell \)-dimensional vector of real uncertain parameters \( q \), each bounded in the interval \([q_i^-, q_i^+]\). More formally, we define

\[
p(s, q) := a_0(q) + a_1(q)s + a_2(q)s^2 + \cdots + a_n(q)s^n
\]

where \( q := (q_1, \ldots, q_\ell) \) and the hypercube \( Q := \{q : q_i^- \leq q_i \leq q_i^+, i = 1, \ldots, \ell\} \) with the nominal parameter \( q^0 \in Q \).

3.1.1 ROBUST \( D \)-STABILITY MARGIN

Definition 3.1 Let \( D \) be an open region in the complex plane and take \( p(s) \) to be a fixed polynomial. Then \( p(s) \) is said to be \( D \)-stable if all its roots lie in the region \( D \).

Definition 3.2 A family of polynomials \( \mathcal{P} = \{p(s, q) : q \in Q\} \) is said to be robustly \( D \)-stable if, for all \( q \in Q \), \( p(s, q) \) is stable; i.e., all roots of \( p(s, q) \) lie in \( D \). For special case when \( D \) is the open left half plane, \( \mathcal{P} \) is simply said to be robustly stable.

Let \( Q \subseteq \mathcal{Q} \). Define value set \( p(z, Q) \subset \mathbb{C} \) by \( p(z, Q) := \{p(z, q) : q \in Q\} \). Define \( kQ := \{k(q - q^0) + q^0 : q \in Q\} \). The Zero Exclusion Condition is stated as follows.

Theorem 3.1 ([24]) The polynomial \( p(s, q) \) is robustly \( D \)-stable for all \( q \in Q \) if and only if \( p(s, q) \) is stable for some \( q \in Q \) and \( 0 \notin p(z, Q) \) for all \( z \in \partial D \).
Let $D_1, D_2, \ldots, D_N$ be disjoint open subsets of the complex plane and suppose $\mathcal{P} = \{p(\cdot, q) : q \in \mathcal{Q}\}$ is a family of polynomials with invariant degree. For each $q \in \mathcal{Q}$ and $i \in \{1, \ldots, N\}$, let $n_i(q)$ denote the number of roots of $p(s, q)$ in $D_i$. Finally, assume that $p(s, q^0)$ has no roots on the boundary of $D = D_1 \cup D_2 \cup \cdots \cup D_N$. Then each of the root indices $n_i(q)$ remains invariant over $\mathcal{Q}$ if and only if the Zero Exclusion Condition $0 \notin p(z, \mathcal{Q})$ is satisfied for all points on $\partial D$.

**Definition 3.3** Suppose $D$ is an open subset of the complex plane with boundary $\partial D$. Then, given an interval $I \subseteq \mathbb{R}$, a mapping $\Phi_D : I \to \partial D$ is said to be a boundary sweeping function for $D$ if $\Phi_D$ is continuous and onto; i.e., $\Phi_D$ is continuous and for each point $z \in \partial D$, there is some $\delta \in I$ such that $\Phi_D(\delta) = z$. The scalar $\delta$ is called a generalized frequency variable for $D$.

Let $k_m(\delta, \mathcal{Q}) := \sup\{k : 0 \notin p(\Phi_D(\delta), k\mathcal{Q})\}$. The robust $D$-stability margin $k_{\text{max}}$ is given by $k_{\text{max}} = \inf_{\delta \in I} k_m(\delta, \mathcal{Q})$. In general, when $D = \bigcup_{l=1}^{N} D_l$ where $D_l$, $l = 1, \ldots, N$ are disjoint open subsets in the complex plane, we can define $N$ boundary sweeping functions $\Phi^l_D : I_l \to \partial D_l$, $l = 1, \ldots, N$ respectively. Then the robust $D$-stability margin is given by

$$k_{\text{max}} = \min_{l=1, \ldots, N} \inf_{\delta \in I_l} k_m(\delta, \mathcal{Q}).$$

### 3.1.2 Domain Splitting Algorithms

It is noted that the analysis of robustness under polynomic structured real parametric uncertainty can be converted into a simpler analysis problem dealing with multilinear structured uncertainty [40, 38].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**Definition 3.4** An uncertain polynomial \( p(s, q) = \sum_{i=0}^{n} a_i(q)s^i \) is said to have a multi-linear uncertainty structure if each of the coefficient functions \( a_i(q) \) is multi-linear. That is, if all but one component of the vector is fixed, then \( a_i(q) \) is affine linear in the remaining component of \( q \). More generally, \( p(s, q) \) is said to have a polynomial uncertainty structure if each of the coefficient functions \( a_i(q) \) is a multi-variable polynomial in the components of \( q \).

In general, there exists no analytic solution for computing exactly \( k_{\max} \). However, the following Mapping Theorem can be applied to obtain a lower bound for \( k_m(\delta, Q) \) for a family of polynomials of multi-linear uncertainty structure.

**Theorem 3.2** ([52]) Suppose an uncertain polynomial \( p(s, q) \) has a multi-linear uncertainty structure. Then

\[
\text{conv } p(z, Q) = \text{conv } \{p(z, q^1), p(z, q^2), \ldots, p(z, q^{2^l})\}, \ \forall z \in \partial \mathcal{D}
\]

where \( \text{conv} \) denotes the convex hull and \( q^1, \ldots, q^{2^l} \) denotes the \( 2^l \) vertices of the hypercube \( Q \).

Let \( k_l(\delta, Q) := \min\{k : 0 \in \text{conv } p(\Phi_D(\delta), kQ)\} \). Then by the Mapping Theorem, \( k_l \) is a lower bound, i.e., \( k_l \leq k_m \).

**Definition 3.5** ([26]) Critical vertices are those adjacent extreme points \( M_\alpha, M_\beta \) of \( \text{conv } p(z, k_iQ) \) such that \( 0 \in \text{conv } \{M_\alpha, M_\beta\} \).

**Definition 3.6** ([26]) \( m(\alpha, \beta) \) is the number of differing coordinates of two vertices \( q^\alpha, q^\beta \) that are mapped by \( p(z, \cdot) \) to \( M_\alpha, M_\beta \), respectively.

It follows from the Mapping Theorem that \( k_l = k_m \) for \( m(\alpha, \beta) = 0, 1 \). For \( m(\alpha, \beta) \geq 2 \), a vertex path is defined as follows.
Definition 3.7 ([26])

A vertex path is any path between critical vertices \( M_\alpha, M_\beta \) consisting of \( m(\alpha, \beta) \) straight-line segments defined by \( p(z, .) \) as \( q \) progresses from \( q^\alpha \) to \( q^\beta \) along the edges of the hypercube \( Q \).

Define

\[
k_u(\delta, Q) := \inf \{ k : \text{At least one of the vertex paths of } \text{conv } p(\Phi_\mathcal{D}(\delta), kQ) \text{ intercepts the origin} \}.
\]

It is shown in [26] that \( k_u \) is an upper bound, i.e., \( k_m \leq k_u \).

In general, it is impractical to compute \( k_{\max} \) over all frequencies. The techniques developed in [26, 38, 40] work essentially as follows.

Choose a range of frequency \( [\delta_l, \delta_u] \subset I \) and grid it as

\[
\delta_j := \frac{(\delta_u - \delta_l)(j - 1)}{n_r n_c}, \quad j = 1, \ldots, n_r n_c \tag{3.1}
\]

where \( n_r \geq 2, n_c \geq 1 \) are integers (In practice, gridding is usually based on the logarithmic scale. However, in this chapter, we use uniform gridding for the simplicity of description.). Apply Algorithm 1 to compute an upper bound \( k^j_u \) and a lower bound \( k^j_l \) for \( k_m(\delta_j, Q) \) such that \( \frac{k^j_l - k^j_l}{k^j_l} < \epsilon, \quad j = 1, \ldots, n_r n_c \). Then an estimate of \( k_{\max} \) can be defined as

\[
\tilde{k}_{\max} := \min_{j=1, \ldots, n_r n_c} k_m(\delta_j, Q)
\]

which satisfies

\[
\min_{j=1, \ldots, n_r n_c} k^j_l \leq \tilde{k}_{\max} \leq \max_{j=1, \ldots, n_r n_c} k^j_u
\]
Algorithm 1 [26] — Computing $k_m(\delta, Q)$

- Step 1: Determine lower bound on $k_m$. Designate the initial uncertain parameter domain, the $n$-dimensional hypercube $Q$, as $Q_{11}$.

- Step 2: Determine upper bound on $k_m$.

- Step 3: Iterate to converge lower and upper bounds to $k_m$. Establish an iterative procedure with counter $r = 1, 2, 3, \ldots$. For each iteration perform the following operations on subdomains $Q_{rp}$ where $p$ represents the number of subdomains left in consideration after the $r$th iteration.

  - Step 3–1: Increment $r$, i.e., $r = r + 1$.

  - Step 3–2: Make orthogonal cuts midway on the longer edges of each subdomain $Q_{rw}$, $w = 1, \cdots, p$ in order that all edge length ratios remain within a factor of 2 of each other. Designate these two subdomains as $Q_{rw}$ and $Q_{r(w+p)}$.

  - Step 3–3: Obtain $k_{lrw} := k_l(\delta, Q_{rw})$ and $k_{lr(w+p)} := k_l(\delta, Q_{r(w+p)})$ via Step 1. (Note: See [26] for handling exceptions).

  - Step 3–4: Obtain $k_{urw} := k_u(\delta, Q_{rw})$ and $k_{ur(w+p)} := k_u(\delta, Q_{r(w+p)})$ via Step 1. (Note: See [26] for handling exceptions).

  - Step 3–5: Repeat Steps 3–2 to 3–4 for each $w = 1, \cdots, p$.

  - Step 3–6: Define

$$ k_{lr} := \min\{k_{lr1}, k_{lr2}, \cdots, k_{lr2p}\} $$
and
\[
k_{u_r} := \min\{k_{u_r1}, k_{u_r2}, \ldots, k_{u_r(2p)}, k_{u_r-1}\}
\]
and \(\epsilon_r = \frac{k_{u_{r-1}} - k_{l_r}}{k_{l_r}}\). (Note: It is shown in [26] that \(k_{l_r-1} < k_{l_r} < k_m \leq k_{u_r} \leq k_{u_{r-1}}\).)

- **Step 3-7:** Eliminate from further consideration all subdomains \(Q_{rw}, w = 1, \cdots, 2p\), whose associated \(k_{l_{rw}} > k_{u_r}\). Designate the number so eliminated as \(u\) and define a new \(p = 2p - u\).

- **Step 3-8:** Repeat Steps 3-1 to 3-8 until \(k_{l_r} \to k_m\). The stop criteria is that \(\epsilon_r\) is less than a chosen tolerance \(\epsilon > 0\).

**Remark 3.1** In the above conventional frequency sweeping algorithm, the most important mechanism which impacts the efficiency is the elimination of subdomains whose lower bound greater than the minimum record of the upper bounds of all subdomains of the frequency being evaluated. This mechanism is implemented in Step 3-7. Since the minimum record of the upper bounds \(k_{u_r}\) is obtained among subdomains for the frequency being evaluated only, the elimination processes for different frequencies are independent.

### 3.2 Domain Splitting and Convergence

One of the most important requirement of an algorithm is its convergence. For example, for the above algorithm, it is expected that given any tolerance \(\epsilon > 0\), the above algorithm stops at finite iteration, i.e., \(r < \infty\) for each frequency. In this section, we investigate how a domain splitting can affect the convergence.
3.2.1 AN IMPRACTICAL ASSUMPTION

The convergence of the above algorithm was addressed in [26, 27] and a Convergence Theorem was proposed. However, in the proof of the Convergence Theorem [26], the convergence was concluded upon the assumption that each subdomain converges to a single point by subdivisions (see [26], lines 40 — 53 of page 162). In another paper [40], the convergence was also concluded by assuming that a subdomain is divided fine enough (see the last paragraph in page 767 of [40]). In fact, such an assumption is in general impractical to be satisfied. This is because the computation is usually very high to divide a domain fine enough to collapse it to a point.

In the above algorithm, the criteria adopted in splitting a domain is to "make orthogonal cut midway on the larger edges". It is also addressed in [27, 40] that a splitting of a domain should be made in a way guaranteeing two critical vertices remained in different subdomains. In general, there is more than one way to satisfy these two criteria. We would like to note that, in general, a splitting scheme which just consists of these two criteria is not sufficient to obtain a sequence of lower bounds (or upper bounds) converging to \( k_m \) or a sequence of subdomains converging to a single point in \( \mathbb{R}^l \).

For example, consider a hypercube \( Q = \{ q \in \mathbb{R}^5 : q_i \in [0, 1], i = 1, \ldots, 5 \} \). Let \( Q_1 = Q \). Based on the above two criteria, \( Q_r \) can be splitted as \( Q_{r+1} \) and \( Q'_{r+1} \) with \( k_m(\delta, Q_{r+1}) = k_m(\delta, Q) \) at the \( r \)-th splitting, \( r = 1, 2, \ldots \). We can not exclude the possibility that there exists \( r_c < \infty \) such that the following are true.

- Critical vertices differ in coordinates \( q_1, q_2, q_3, q_4 \) for \( r > r_c \).
- Coordinates \( q_1, q_2 \) are cut in round robin order for \( r > r_c \).
Finally, we will end up with a degenerate hypercube, with $q_1$, $q_2$, $q_5$ being constants and $q_3$, $q_4$ varying within intervals, i.e., a planar "box". Because $q_3$, $q_4$ can vary in intervals, it is possible that there is a gap between the upper bound $k_u$ and the lower bound $k_l$, i.e., $\exists \nu > 0$ such that $k_u - k_l > \nu$.

Therefore, it is important to raise the following question:

*What kind of splitting guarantees the convergence?*

### 3.2.2 Guaranteed Critical Vertices Distance Convergence

Consider a hypercube $Q = \{ q \in \mathbb{R}^\ell : q_i \in [q_i^-, q_i^+], i = 1, \ldots, \ell \}$. Define a sequence (finite or infinite) of domains $\{Q_r\}$ iteratively as follows.

- **Step 1:** Let $Q_1 = Q$. Let $r = 1$.

- **Step 2:** If the critical vertices of domain $Q_r$, denoted by $q_{i}^a$ and $q_{i}^b$, differ in no more than one coordinates then the iteration process is terminated, otherwise choose $i^* \in \mathcal{U}_r := \{ i : q_{i}^{a_r} \neq q_{i}^{b_r} \}$ and designate either $\{ q \in Q_r : q_{i}^{a_r} + q_{i}^{b_r} \leq q_i \leq \max\{q_{i}^{a_r} + q_{i}^{b_r}, q_{i}^{a_r}, q_{i}^{b_r}\} \}$ or $\{ q \in Q_r : \min\{q_{i}^{a_r}, q_{i}^{b_r}\} \leq q_i \leq \frac{q_{i}^{a_r} + q_{i}^{b_r}}{2} \}$ as $Q_{r+1}$.

- **Step 3:** Set $r = r + 1$. Go to Step 2.

In general, there are more than one way to choose $i^* \in \mathcal{U}_r$. Let $Q_r = \{ q \in \mathbb{R}^\ell : q_i \in [q_i^{a_r}, q_i^{b_r}], i = 1, \ldots, \ell \}$, we can define a splitting scheme as follows.

**Definition 3.8** A maximal-cut is a partition of $Q_r$ as above by choosing $i^* \in \mathcal{U}_r$ such that

$$q_{i^*,r}^+ - q_{i^*,r}^- = \max_{i \in \mathcal{U}_r} q_{i,r}^+ - q_{i,r}^-.$$
Another splitting scheme adopted in [27] was that the cut should be made over the coordinate that has been subdivided the least number of times. More formally, we define a fair-cut scheme as follows.

**Definition 3.9** A fair-cut is a partition of $Q_r$ as above by choosing $i_\ast \in \mathcal{U}_r$ such that

$$\frac{q_{i_\ast}^- - q_{i_\ast}^+}{q_{i_\ast}^+ - q_{i_\ast}^-} = \min_{i \in \mathcal{U}_r} \frac{q_i^+ - q_i^-}{q_i^+ - q_i^-}.$$

Now we discuss the properties of the above two domain splitting schemes.

**Theorem 3.3** Let $\{Q_r\}$ be a sequence of domains generated as above by applying the maximal-cut scheme in each splitting. Then, we have that either $\{Q_r\}$ is a finite sequence, i.e., $\exists r_0 < \infty$ such that the critical vertices of $Q_{r_0}$ differ in no more than one coordinates, or $\{Q_r\}$ is an infinite sequence such that

$$\lim_{r \to \infty} ||p(\Phi_D(\delta), q^{\alpha,r}) - p(\Phi_D(\delta), q^{\beta,r})|| = 0$$

and

$$\lim_{r \to \infty} k_l(\delta, Q_r) = \lim_{r \to \infty} k_u(\delta, Q_r) = \lim_{r \to \infty} k_m(\delta, Q_r).$$

Moreover, the same result follows if $\{Q_r\}$ is a sequence of domains generated as above by applying the fair-cut scheme in each splitting.

**Proof.** We only need to consider the case that $\{Q_r\}$ is an infinite sequence. Decompose the coordinates index set $\mathcal{I} = \{1, \ldots, \ell\}$ as $\mathcal{I} = \mathcal{I}_f \cup \mathcal{I}_\infty$ where

$$\mathcal{I}_f = \{i \in \mathcal{I} : [q_i^-, q_i^+] \text{ is divided finite many times}\}$$
and

\[ \mathcal{I}_\infty = \{ i \in \mathcal{I} : [q_i^-, q_i^+] \text{ is divided infinite many times}\}. \]

Obviously, \( \lim_{r \to -\infty} ||q^r - q^0|| = 0 \) for the case that \( \mathcal{I}_f = \emptyset \). We only need to consider the case that \( \mathcal{I}_f \neq \emptyset, \mathcal{I}_\infty \neq \emptyset \). Note that \( \exists r_1 > 0 \) such that \( q_{i,r}^+ = q_{i,r_1}^+, \ q_{i,r}^- = q_{i,r_1}^- \), \( \forall i \in \mathcal{I}_f, \forall r \geq r_1 \). Define

\[ \zeta = \min_{i \in \mathcal{I}_f} q_{i,r_1}^+ - q_{i,r_1}^- \]

Then

\[ \min_{i \in \mathcal{I}_f} q_{i,r}^+ - q_{i,r}^- = \zeta > 0, \forall r \geq r_1. \]

Note that \( \exists r_2 > 0 \) such that

\[ q_{i,r}^+ - q_{i,r}^- < \zeta, \forall i \in \mathcal{I}_\infty, \forall r \geq r_2. \]

We claim that \( \mathcal{U}_r \cap \mathcal{I}_f = \emptyset, \forall r > \max\{r_1, r_2\} \). In fact, if this is not the case, then \( \exists i_* \in \mathcal{U}_r \cap \mathcal{I}_f \) such that

\[ q_{i_*,r}^+ - q_{i_*,r}^- = \max_{i \in \mathcal{U}_r} q_{i,r}^+ - q_{i,r}^- \geq \zeta \]

because \( q_{i,r}^+ - q_{i,r}^- < \zeta, \forall i \in \mathcal{U}_r \cap \mathcal{I}_\infty \). It follows that \( Q_r \) is splitted as \( Q_{r+1} \) and \( Q'_{r+1} \) by dividing interval \([q_{i_*,r}^-, q_{i_*,r}^+]\), which contradicts to \( q_{i,r}^+ = q_{i,r_1}^+, \ q_{i,r}^- = q_{i,r_1}^- \), \( \forall i \in \mathcal{I}_f, \forall r \geq r_1 \). Thus, the claim is true and it follows that

\[ ||q^r - q^0||^2 = \sum_{i \in \mathcal{I}_\infty} (q_{i,r}^a - q_{i}^0)^2 \leq \sum_{i \in \mathcal{I}_\infty} (q_{i,r_1}^+ - q_{i,r_1}^-)^2, \forall r > \max\{r_1, r_2\}. \]
Therefore, \( \lim_{r \to \infty} \| q^{a_r} - q^{b_r} \| = 0 \). Since \( p(z, q) \) is a continuous function of \( q \), it follows that \( \lim_{r \to \infty} \| p(\Phi_D(\delta), q^{a_r}) - p(\Phi_D(\delta), q^{b_r}) \| = 0 \). By the definition of \( k_l \) and \( k_u \), we have

\[
\lim_{r \to \infty} k_l(\delta, Q_r) = \lim_{r \to \infty} k_u(\delta, Q_r) = \lim_{r \to \infty} k_m(\delta, Q_r).
\]

Similarly, to show that the same result follows if \( \{ Q_r \} \) is a sequence of domains generated as above by applying the fair-cut scheme in each splitting, we only need to consider the case that \( \mathcal{I}_f \neq \emptyset \), \( \mathcal{I}_\infty \neq \emptyset \). Note that \( \exists r_3 > 0 \) such that \( q^+_{i,r} = q^+_{i,r_3}, \ q^-_{i,r} = q^-_{i,r_3}, \ \forall i \in \mathcal{I}_f, \ \forall r \geq r_3 \). Define

\[
n_s = \max_{i \in \mathcal{I}_f} \frac{q^+_i - q^-_i}{q^+_{i,r_3} - q^-_{i,r_3}}.
\]

Then

\[
\max_{i \in \mathcal{I}_f} \frac{q^+_i - q^-_i}{q^+_{i,r} - q^-_{i,r}} = n_s < \infty, \ \forall r \geq r_3.
\]

Note that \( \exists r_4 > 0 \) such that

\[
\frac{q^+_i - q^-_i}{q^+_{i,r} - q^-_{i,r}} > n_s, \ \forall i \in \mathcal{I}_\infty, \ \forall r \geq r_4.
\]

We claim that \( \mathcal{U}_r \cap \mathcal{I}_f = \emptyset, \ \forall r > \max\{r_3, r_4\} \). In fact, if this is not the case, then \( \exists i_\ast \in \mathcal{U}_r \cap \mathcal{I}_f \) such that

\[
\frac{q^+_i - q^-_i}{q^+_{i,r} - q^-_{i,r}} = \min_{i \in \mathcal{U}_r} \frac{q^+_i - q^-_i}{q^+_{i,r} - q^-_{i,r}} \leq n_s,
\]

because \( \frac{q^+_i - q^-_i}{q^+_{i,r} - q^-_{i,r}} > n_s, \ \forall i \in \mathcal{U}_r \cap \mathcal{I}_\infty \). It follows that \( Q_r \) is splitted as \( Q_{r+1} \) and \( Q'_{r+1} \) by dividing interval \( [q^-_{i,r}, q^+_{i,r}] \), which contradicts to \( q^+_{i,r} = q^+_{i,r_3}, \ q^-_{i,r} = q^-_{i,r_3}, \ \forall i \in \mathcal{I}_f, \ \forall r \geq r_3 \).
Thus, the claim is true and it follows that

\[ ||q^{\alpha} - q^{\beta}||^2 = \sum_{i \in I_f} (q_i^{\alpha} - q_i^{\beta})^2 \leq \sum_{i \in I_f} (q_{i,r-1}^+ - q_{i,r-1}^-)^2, \quad \forall r > \max\{r_3, r_4\}. \]

Therefore, by the same argument as in the maximal-cut schemes, the result follows.

\[ \square \]

Remark 3.2 From the proof of the theorem we can see that both domain splitting schemes guarantee \( ||q^{\alpha} - q^{\beta}|| \to 0 \) while allow \( Q_r \to Q_\infty \) where \( Q_\infty \) is not a single point in \( \mathbb{R}^4 \). Clearly, to make a subdomain converge to a single point requires much more computational effort than to make \( ||q^{\alpha} - q^{\beta}|| \to 0 \). As we can see later, \( ||q^{\alpha} - q^{\beta}|| \to 0 \) leads to the existence of a sequence of lower bounds (or upper bounds) converging to \( k_m \). Therefore, an algorithm based on the maximal-cut (or fair-cut) splitting scheme will reduce much computational effort in computing \( k_m \) than other algorithms based on making subdomains converge to a single point in \( \mathbb{R}^4 \).

From the proof, we can also see that the convergence will not follow if the domain splitting is made along the larger but not the largest edges of each subdomain. It was remarked in [27] that a fair-cut avoids the problem of getting into very narrow and long subdomains which can decrease the convergence speed. From the proof, we can see that it plays a role much more than affecting the speed of convergence. It is a sufficient condition to the existence of a sequence of lower bounds (or upper bounds) converging to \( k_m \). We would like to point out that the maximal-cut scheme has better worst case convergence behavior than that of the fair-cut scheme.

To see the efficiency of the maximal-cut (or the fair-cut) splitting scheme, it is helpful to compare the image of the last subdomain resulted from the the maximal-
cut (or the fair-cut) splitting scheme and the image of the last subdomain obtained by the finely subdivision. The situation is shown in the following Figure 3.2.

![Image of the last subdomain](image1)

**Figure 3.2: The Image of the Last Subdomain**

### 3.3 Parallel Frequency Sweeping Algorithm

In this section we shall investigate a new frequency sweeping structure.

#### 3.3.1 The Main Root of Inefficiency

To the best of our knowledge, no effort in the existing literature has been devoted to exploit a smart frequency sweeping strategy. Existing techniques are basically as follows.

Choose and grid a range of frequency. Then calculate exactly \( k_m \) for each grided frequency. Finally, compare to find the minimum \( k_m \) and return it as an estimate of \( k_{\text{max}} \).

For complicated root region \( \mathcal{D} \), the number of frequencies to be evaluated for \( k_m \) would be substantial in order to obtain a reasonably good estimate for \( k_{\text{max}} \). Even
if the computation of $k_m$ for each frequency is very efficient, the overall complexity is still very high, because we need to evaluate $k_m$ for many frequencies.

Thus for the sake of efficiency, it is natural to conceive a smart frequency sweeping strategy. More specifically, we would raise the following question,

*Is it possible to obtain the stability margin $k_{\text{max}}$ without tightly bounding $k_m(\delta_j, \mathcal{Q})$ for each frequency $\delta_j$?*

The following section is devoted to answering this question.

### 3.3.2 Parallel Frequency Sweeping Algorithm

Consider the same set of grided frequencies $\delta_j, \ j = 1, \cdots, n_r n_c$ defined by (3.1) and relabel them as

$$
\delta_{ij} := \delta_l + \frac{(\delta_u - \delta_l)(i - 1)}{n_r} + \frac{(\delta_u - \delta_l)(j - 1)}{n_r n_c}, \ i = 1, \cdots, n_r, \ j = 1, \cdots, n_c.
$$

We are now in a position to present our Parallel Frequency Sweeping Algorithm as follows.

**Algorithm 2 — Parallel Frequency Sweeping Algorithm**

- Step 1: Initialize. Set $j = 1$. Set $\hat{k} = \infty$. Set tolerance $\epsilon > 0$. Set maximal iteration number $\mathit{IT}$.
- Step 2: Update $\hat{k}$ and record the number of iterations $r(j)$ for frequency $\delta_{ij}$ by the following steps.
  - Step 2-1: Let $\mathcal{U}_{ij} = \{Q_k\} = \mathcal{Q}, \ i = 1, \cdots, n_r$. Set $r = 1$.
  - Step 2-2: If $r = \mathit{IT} + 1$ or $\mathcal{U}_{ij}$ is empty for any $i \in \{1, \cdots, n_r\}$ then record $r(j) = r$ and go to Step 3, else do the following for all $i$ such that $\mathcal{U}_{ij}$ is not empty.
* Choose \( Q \) to be any element of \( \mathcal{U}_{ij} \) with

\[
k_l(\delta_{ij}, Q) = \min_{Q_k \in \mathcal{U}_{ij}} k_l(\delta_{ij}, Q_k).
\]

* Partition \( Q \) into \( Q_a \) and \( Q_b \) by applying a maximal-cut.

* Remove \( Q \) from \( \mathcal{U}_{ij} \).

* Update

\[
\hat{k} = \min\{\hat{k}, \ k_u(\delta_{ij}, Q_a), \ k_u(\delta_{ij}, Q_b)\}. \tag{3.2}
\]

* Add any \( Q \in \{Q_a, Q_b\} \) with two or more critical vertices to \( \mathcal{U}_{ij} \).

* Remove from \( \mathcal{U}_{ij} \) any \( Q \) with

\[
0 \notin \operatorname{conv}\left(p\left(\Phi_D(\delta_{ij}), \frac{\hat{k}}{1 + \epsilon} - Q\right)\right). \tag{3.3}
\]

- Step 2–3: Set \( r = r + 1 \) and go to Step 2–2.

• Step 3: If \( j = n_c \) then STOP, else set \( j = j + 1 \) and go to Step 2.

In Algorithm 2, \( n_r \) branches of frequency sweeping are performed in parallel with starting frequencies \( \delta_{i1}, \ i = 1, \ldots, n_r \) and step size \( \frac{\delta_{i0} - \delta_{i1}}{n_r n_c} \). Each branch of frequency sweeping is not independent, they exchange information. The information is applied to determine the subdomains to be eliminated from further consideration and to update \( \hat{k} \). Finally, \( \hat{k} \) is returned as the robust stability margin. Algorithm 2 is visualized in the following Figure 3.3 for the case of \( N = 3, \ K = 4 \).

**Remark 3.3** As we can see from Step 2–2, there are two mechanisms contribute to the efficiency of the Parallel Frequency Sweeping Algorithm. First, any subdomain...
$Q$ satisfies

$$0 \notin \text{conv}(p \left( p_{\Phi}(\delta_{ij}), \frac{\hat{k}}{1+\epsilon} Q \right)) \iff k_i(\delta_{ij}, Q) < \frac{\hat{k}}{1+\epsilon}$$

will never be partitioned again and thus can be eliminated from further consideration. Second, any subdomain $Q$ with critical vertices differing in no more than one coordinates will never be partitioned again and thus can be eliminated from further consideration.

We would like to note that the proposed Parallel Frequency Sweeping provides substantial improvement on efficiency over the algorithms in [26]. This can be explained by the significant relaxation of the condition for eliminating a subdomain from consideration. By (3.2) and (3.4) we can see that $\hat{k}$ is the minimum record of the upper bounds among all subdomains evaluated (no matter belong to the same frequency or not) and is contracted to $\frac{\hat{k}}{1+\epsilon}$. In contrast, in algorithms of [26] the minimum record of upper bounds is obtained for the frequency being considered only.
Therefore, the condition for eliminating a subdomain from consideration is much looser than its counterpart of algorithms in [26]. Consequently, such a significant relaxation results in a substantial decrease of the number of subdomains needed to be evaluated.

Remark 3.4 In Algorithm 2, at each iteration, only the domain with the smallest lower bound is partitioned. This mechanism differs from that of Algorithm 1 in which all domains are partitioned and thus the number of domains increases exponentially. We can see that Algorithm 2 effectively controls the explosion of the number of domains and thus is much efficient than the conventional algorithm.

Remark 3.5 It is important to note that Algorithm 2 involves only one CPU processor. It is fundamentally different from the parallel algorithms which involves more than one CPU processors.

In addition to the novel frequency sweeping strategy, another character of Algorithm 2 is that there is no tolerance criteria directly forced on the final result, however, the final result falls into tolerance automatically.

Theorem 3.4 Suppose that the maximum iteration number IT = ∞. For arbitrary tolerance ϵ > 0, Algorithm 2 stops with a finite number of domain splittings for each j, i.e., r(j) < ∞, ∀j. Moreover, the final \( \hat{k} \) satisfies

\[
0 \leq \frac{\hat{k} - \bar{k}_{max}}{\bar{k}_{max}} < \epsilon.
\]

Proof. We first show the final \( \hat{k} \geq \bar{k}_{max} \). Let \( k_u \) be the upper bound of domain \( Q \subseteq Q \) which ever appeared during the execution of Algorithm 2. Let \( \delta \) be the
associated frequency of $Q$. Note that $0 \in p(\delta, k_u Q) \subset p(\delta, k_u Q)$ and thus $k_u \geq \tilde{k}_{\text{max}}$. Note that the final $\tilde{k}$ is the minimum record of all such $k_u$'s, thus $\tilde{k} \geq \tilde{k}_{\text{max}}$.

We next need to show that Algorithm 2 stops with a finite iteration number $r(j)$ for each $j$. Suppose $\exists j$ such that $r(j)$ goes to $\infty$. Then $\exists \delta_{ij}$ such that $r(j)$ goes to $\infty$. Therefore, we can construct a sequence of nested domains $\{Q_r\}$ such that $Q_1 \supset Q_2 \supset \cdots \supset Q_r \supset Q_{r+1} \supset \cdots$. Thus by Theorem 3.3 we have that $\forall \varepsilon > 0$, $\exists r_0 < \infty$ such that $k_u(\delta_{ij}, Q_r) - k_l(\delta_{ij}, Q_r) < \frac{\varepsilon}{1 + \varepsilon} \tilde{k}_{\text{max}}$, $\forall r \geq r_0$. Thus $\min\{\hat{k}, k_u(\delta_{ij}, Q_{r_0})\} - k_l(\delta_{ij}, Q_{r_0}) < \frac{\varepsilon}{1 + \varepsilon} \tilde{k}_{\text{max}}$. Note that $k_l(\delta_{ij}, Q_{r_0+1}) \geq k_l(\delta_{ij}, Q_{r_0})$ because $Q_{r_0+1} \subset Q_{r_0}$. Also note that $\hat{k}$ never increases, thus we have $\hat{k} - k_l(\delta_{ij}, Q_{r_0+1}) < \frac{\varepsilon}{1 + \varepsilon} \tilde{k}_{\text{max}}$. Note that $\hat{k} \geq \tilde{k}_{\text{max}}$, we have

$$\hat{k} - k_l(\delta_{ij}, Q_{r_0+1}) < \frac{\varepsilon}{1 + \varepsilon} \tilde{k}_{\text{max}} \implies \frac{\hat{k}}{1 + \varepsilon} < k_l(\delta_{ij}, Q_{r_0+1}),$$

which implies that $0 \notin \text{conv} \left( p\left( \Phi_D(\delta_{ij}), \frac{\hat{k}}{1 + \varepsilon} Q_{r_0+1} \right) \right)$. Therefore, by Algorithm 2 $Q_{r_0+1}$ will not be split. This is a contradiction. So, we have shown that Algorithm 2 stops with a finite number of domain splitings for each $j$.

Note that $\exists \delta_{ij}$ such that $k_m(\delta_{ij}, Q) = \tilde{k}_{\text{max}}$. Moreover, $\exists q^* \in Q$ such that $p(\delta_{ij}, \tilde{k}_{\text{max}} q^*) = 0$. Since Algorithm 2 stops with a finite number of domain splitings for each $j$, we have that all subdomains ever generated are finally eliminated from consideration. Thus, there must exist $Q^*$ which contains $q^*$ be eliminated from consideration at a certain level of spliting.

Assume that $\hat{k} = \tilde{k}$ when $Q^*$ is eliminated from consideration. Then either $0 \notin p(\delta_{ij}, \frac{\hat{k}}{1 + \varepsilon} Q^*)$ or the critical vertices of $Q^*$ differ in no more than one coordinates. If the first case is true, then by $p(\delta_{ij}, \tilde{k}_{\text{max}} q^*) = 0$ and $q^* \in Q^*$, we have that $0 \in p(\delta_{ij}, \tilde{k}_{\text{max}} Q^*)$. Thus by $0 \notin p(\delta_{ij}, \frac{\hat{k}}{1 + \varepsilon} Q^*)$, we have $\frac{\hat{k}}{1 + \varepsilon} < \tilde{k}_{\text{max}}$. Obviously, the
final $\hat{k} \leq \bar{k}$ and thus
\[
\frac{\hat{k}}{1 + \epsilon} < \bar{k}_{\text{max}} \implies \frac{\hat{k} - \bar{k}_{\text{max}}}{\bar{k}_{\text{max}}} < \epsilon.
\]

If the latter case is true, then we have
\[
k_i(\delta_{ij}, Q^*) = k_u(\delta_{ij}, Q^*) = \bar{k}_{\text{max}} = \hat{k}.
\]

The proof is thus completed.

Remark 3.6 From the proof, we can see that the existence of a sequence of upper bounds converging to $\bar{k}_{\text{max}}$ is due to the convergence of the distance of critical vertices.

Theorem 3.5 Suppose that the maximum iteration number $IT < \infty$ and that Algorithm 2 stops with $r(j) < IT, \forall j$. Then the final $\hat{k}$ satisfies
\[
0 \leq \frac{\hat{k} - \bar{k}_{\text{max}}}{\bar{k}_{\text{max}}} < \epsilon.
\]

Proof. Since Algorithm 2 stops with $r(j) < IT, \forall j$, we can conclude that all subdomains ever generated are finally eliminated from consideration. Thus the result follows from similar argument as for Theorem 3.4.
3.4 An Illustrative Example

Our computational experience shows that Algorithm 2 provides a significant improvement upon conventional algorithms for most control problems. Moreover, the improvement depends on the problems and can be arbitrarily good.

Consider a linear system as follows:

\[ \dot{x} = A(q)x + Bu \]

\[ y = Cx \]

where

\[
B = \begin{bmatrix}
1 \\
2 \\
1 \\
1 \\
\end{bmatrix}, \quad C = \begin{bmatrix}
1 & 0.5 & 1 & 0.5
\end{bmatrix}
\]

and

\[
A(q) = \begin{bmatrix}
-1 - 0.5q_1 & -10 & -1 & 10 \\
-0.5 & -1 & 1 & 0.5 \\
0.5 & -4 & -1 & -10 \\
-10 & 0.5 & 0 & -2.5 - 1.5q_2
\end{bmatrix}
\]
with uncertain parameter $q \in \mathcal{Q} = [0, 1] \times [0, 3] \subset \mathbb{R}^2$. We obtained a polynomial for this system as $p(s, q) = \text{det}(sI - A(q))$ which has a multilinear structure.

The upper bound and lower bound of $k_m$ on $\mathcal{Q}$ is shown in Figure 3.4, 3.5, 3.6 and 3.7. We can see that for most of the frequencies the upper bound and lower bound are far apart and thus the importance of domain splitting is obvious.

To compute $k_{\text{max}}$, we uniformly grid frequency band $[0.01, 15.01]$ and obtain 1,500 grid frequencies as

$$\omega_j = 0.01j, \quad j = 1, \cdots, 1500.$$ 

In Algorithm 2, we choose the relative error $\epsilon = 0.01$ and $n_r = 30$, $n_c = 50$. The 1,500 frequencies are regrouped as

$$\omega_{ij} = 0.01 + 0.2(i - 1) + 0.01(j - 1), \quad i = 1, \cdots, 30; \quad j = 1, \cdots, 50.$$

We ran the program in Sun Spark 10 workstation. The running time is about 80 seconds. The total number of domains evaluated is 1,570. We obtained $\hat{k} = 1.4384$ which is achieved at frequency $\omega_{20,18} = 9.78$. By Theorem 3.4, we can concluded that

$$0 \leq \frac{\hat{k} - \bar{k}_{\text{max}}}{\bar{k}_{\text{max}}} < \epsilon = 0.01.$$

To compare the performance of conventional algorithm with that of Algorithm 2, it is fair to choose the tolerance $\varepsilon_r = 0.01$ in Algorithm 1. We also ran the program in Sun Spark 10 workstation. The running time is about 9 hours. The total number of domains evaluated is 64,813. We obtained $\hat{k} = 1.4380$ which is achieved at frequency $\omega_{9783} = 9.783$. Therefore, Algorithm 2 has a speed-up of 400 over the
conventional algorithm. Moreover, the number of domains evaluated in Algorithm 2 is only a small fraction (which is $\frac{1570}{64813} \approx 0.0242$) of that of the conventional algorithm. The number of domains evaluated in Algorithm 2 and the conventional one for each frequency is shown respectively in Figure 3.8 and Figure 3.9.

We can see that Algorithm 2 provides much superior performance than the conventional algorithms. The improvement comes from the characteristic eliminating mechanisms in Algorithm 2. More formally, we describe the eliminating process in Algorithm 2 as follows.

Let $U$ be a record of global upper bound achieved by frequency $\omega$. Let $Q_{ij} \subseteq Q$ be a domain associated with frequency $\omega_{ij}$. When $Q_{ij}$ is eliminated, i.e., $k_t(\omega_{ij}, Q_{ij}) > \frac{U}{1+\epsilon}$ is satisfied, there are only three cases as follows.

- Case (i): $\omega_{ij} < \omega$. We call the elimination as *Backward Pruning*.
- Case (ii): $\omega_{ij} > \omega$. We call the elimination as *Forward Pruning*.
- Case (iii): $\omega_{ij} = \omega$. We call the elimination as *Present Pruning*.

All the above three types of pruning processes play important roles in Algorithm 2. However, there is only *Present Pruning* in the conventional algorithm. Therefore, Algorithm 2 has a much powerful pruning mechanism and is much more efficient.

In this example, we have 24 records which are shown in Figure 3.10 and Figure 3.11, where the horizontal coordinate represents the record value and the vertical coordinate represents the frequency achieving it. Two consequent records are connected by dashed line. The effectiveness of the three types of pruning processes are shown respectively in Figure 3.12, 3.13 and 3.14, where the vertical coordinates represents the number of domains eliminated by the record and the horizontal coordinate represents the record index.
Figure 3.4: \( k_m \) Upper Bound (solid) and Lower Bound (dashed) on \( Q \)

Figure 3.5: \( k_m \) Upper Bound (solid) and Lower Bound (dashed) on \( Q \)
Figure 3.6: $k_m$ Upper Bound (solid) and Lower Bound (dashed) on $Q$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 3.8: Domains Evaluated in Algorithm 2

Figure 3.9: Domains Evaluated in the Conventional Algorithm
Figure 3.10: Evolution of the Global Upper Bound

Figure 3.11: Evolution of the Global Upper Bound
Figure 3.12: Backward Pruning

Figure 3.13: Forward Pruning
Figure 3.14: Present Pruning
CHAPTER 4

ORDER STATISTICS AND PROBABILISTIC ROBUST CONTROL

In Chapter 2 and Chapter 3 we have developed deterministic algorithms for robustness analysis problems. Although we have achieved significant improvements in efficiency, the problems of NP hardness and conservativeness are remained unsolved. In light of this situation, we study the robust control problems in a probabilistic framework.

In this chapter order statistics theory is applied to probabilistic robust control theory to compute the minimum sample size needed to come up with a reliable estimate of an uncertain quantity under continuity assumption of the related probability distribution. Also, the concept of distribution-free tolerance intervals is applied to estimate the range of an uncertain quantity and extract the information about its distribution. To overcome the limitations imposed by the continuity assumption in the existing order statistics theory, we have derived a cumulative distribution function of the order statistics without the continuity assumption and developed an inequality showing that this distribution has an upper bound which equals to the corresponding distribution when the continuity assumption is satisfied. By applying this inequality, we investigate the minimum computational effort needed to come up with an reliable estimate for the upper bound (or lower bound) and the range of a quantity. We also give conditions, which are much weaker than the absolute continuity assumption, for the existence of such minimum sample size. Furthermore, the issue of making tradeoff between performance level and risk is addressed and
This chapter is organized as follows. Section 4.1 presents the problem formulation and motivations. In Section 4.2, we derive a distribution inequality without the continuity assumption. Design tradeoff is discussed in Section 4.3 using the distribution derived in Section 4.2. Section 4.4 gives the minimum sample size under various assumptions and Section 4.5 considers the tolerance intervals.

4.1 Problem Formulation and Motivations

Let \( q \) be a random vector, bounded in a compact set \( \mathcal{Q} \), with a multivariate probability density function \( \varpi(q) \). Let \( u(q) \) be a real scalar measurable function of the random vector \( q \) with cumulative probability distribution \( F_u(\gamma) := \Pr \{ u(q) \leq \gamma \} \) for a given \( \gamma \in \mathbb{R} \). Let \( q^1, q^2, \ldots, q^N \) be the i.i.d. (independent and identically distributed) samples of \( q \) generated according to the same probability density function \( \varpi(q) \) where \( N \) is the sample size. Now define random variable \( \hat{u}_i, i = 1, 2, \ldots, N \) as the \( i \)-th smallest observation of \( u(q) \) during \( N \) sample experiments. These random variables are called order statistics \([1, 20]\) because \( \hat{u}_1 \leq \hat{u}_2 \leq \hat{u}_3 \leq \cdots \leq \hat{u}_N \).

We are interested in computing the following probabilities:

- \( \Pr \{ \Pr \{ u(q) > \hat{u}_n \} \leq \epsilon \} \) for any \( 1 \leq n \leq N \), and \( \epsilon \in (0, 1) \);
- \( \Pr \{ \Pr \{ u(q) < \hat{u}_m \} \leq \epsilon \} \) for any \( 1 \leq m \leq N \), and \( \epsilon \in (0, 1) \);
- \( \Pr \{ \Pr \{ \hat{u}_m < u(q) \leq \hat{u}_n \} \geq 1 - \epsilon \} \) for any \( 1 \leq m < n \leq N \), and \( \epsilon \in (0, 1) \).

In the subsequent subsections, we shall give some motivations for computing such probabilities.
4.1.1 Robust Analysis and Optimal Synthesis

As noted in [33] and [45], to tackle robust analysis or optimal synthesis problem, it is essential to deal with the following questions:

- What is $\max_Q \ u(q)$ (or $\min_Q \ u(q)$)?

- What is the value of $q$ at which $u(q)$ achieves $\min_Q \ u(q)$ (or $\max_Q \ u(q)$)?

Consider, for example, an uncertain system shown in Figure 4.1. Denote the transfer function from $v$ to $z$ by $T_{xv}$ and suppose that $T_{xv}$ has the following state space realization

$$
T_{xv} = \begin{bmatrix}
A(q) & B(q) \\
C(q) & D(q)
\end{bmatrix}.
$$

We can now consider following robustness problems:

- Robust stability: Let $u(q) := \max_i \text{Re} \lambda_i(A(q))$ where $\lambda_i(A)$ denotes the $i$-th eigenvalue of $A$. Then the system is robustly stable if $\max_{q \in Q} \ u(q) < 0$.

- Robust performance: Suppose $A(q)$ is stable for all $q \in Q$. Define $u(q) := ||T_{xv}||_{\infty}$. Then the robust performance problem is to determine if $\max_{q \in Q} \ u(q) \leq \gamma$ is satisfied for some prespecified $\gamma > 0$. 

Figure 4.1: Uncertain System
As another example, consider a dynamical system shown in Figure 4.2 and suppose $q$ is a vector of controller parameters to be designed. Denote the transfer function from $d$ to $z$ by $T_{zd}$ and suppose $T_{zd}$ has the following state space realization

$$T_{zd} = \begin{bmatrix} A_s(q) & B_s(q) \\ C_s(q) & D_s(q) \end{bmatrix}.$$ 

Let $u(q) := ||T_{zd}||_\infty$ and $Q_s = \{q \in Q : A_s(q) \text{ is stable}\}$. Then an $H_\infty$ optimal design problem is to determine a vector of parameters $q$ achieving $\min_{Q_s} u(q)$.

In general, exactly evaluating $\min_{Q_s} u(q)$ (or $\max_{Q_s} u(q)$) or determining $q$ achieving it may be an NP hard problem and thus is intractable in practice. Henceforth, we adopt the probabilistic approach proposed in [33] and [45]. That is, estimating $\min_{Q_s} u(q)$ as

$$\hat{u}_1 = \min_{i \in \{1, 2, \ldots, N\}} \hat{u}_i$$

for sufficiently large $N$ and computing $\Pr \{\Pr \{u(q) < \hat{u}_1\} \leq \varepsilon\}$ for a small $\varepsilon \in (0, 1)$ to see how reliable the estimation is. Similarly, we estimate $\max_{Q_s} u(q)$ as $\hat{u}_N = \max_{i \in \{1, 2, \ldots, N\}} \hat{u}_i$ for sufficiently large $N$ and consider $\Pr \{\Pr \{u(q) > \hat{u}_N\} \leq \varepsilon\}$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.1.2 Quantity Range

In many applications, estimating only the upper bound (or lower bound) for a quantity is not sufficient. It is also important to estimate the range of the quantity with a certain accuracy and confidence level. For example, in pole placement problem, we need to know the range which the poles fall into. Suppose that \( q \) is the vector of uncertain parameters or design parameters of a system and \( u(q) \) is an uncertain quantity, for example, \( u(q) \) may be the \( H_\infty \) norm of a closed-loop transfer function or the maximum real part of the eigenvalues of the closed-loop system matrix. Intuitively, the range of quantity \( u(q) \) can be approached by \( (\hat{u}_1, \hat{u}_N) \) as sample size \( N \) goes to infinity. Therefore, it is important to know \( \Pr\{\Pr\{u(q) < u(q) \leq \hat{u}_N\} \geq 1 - \epsilon\} \).

So far, we have only concerned the lower bound and (or) upper bound of uncertain quantity \( u(q) \). Actually, it is desirable to know its distribution function \( F_u(.) \). This is because \( F_u(.) \) contains all the information of the quantity. However, the exact computation of \( F_u(.) \) is, in general, intractable [51]. An alternative is to extract as much as possible the information of the distribution function \( F_u(.) \) from the observations \( \hat{u}_i, \ i = 1, 2, \cdots, N \). For this purpose, we are interested in computing the probabilities asked at the beginning of this section. In particular, we will see in section 4.4 that, computing \( \Pr\{\Pr\{u(q) \geq \hat{u}_n\} \leq \epsilon\} \) is of great importance to make the tradeoff between the performance gradation and risk when designing a controller.

4.2 Distribution Inequality

Note that \( \Pr\{u(q) \leq \hat{u}_i\} = F_u(\hat{u}_i), \ i = 1, 2, \cdots, N \). It is important to know the associated distribution of any \( k \) random variables \( F_u(\hat{u}_{i_1}), F_u(\hat{u}_{i_2}), \cdots, F_u(\hat{u}_{i_k}), \ 1 \leq
\[ i_1 < i_2 < \cdots < i_k \leq N, \quad 1 \leq k \leq N, \text{ i.e.,} \]
\[ F(t_1, t_2, \ldots, t_k) := \Pr \{ F_u(\hat{u}_{i_1}) \leq t_1, F_u(\hat{u}_{i_2}) \leq t_2, \ldots, F_u(\hat{u}_{i_k}) \leq t_k \}. \]

To that end, we have the following theorem which follows essentially by combining Probability Integral Transformation Theorem in [29] and Theorem 2.2.3 in [20].

**Theorem 4.1** Let \( 0 = t_0 \leq t_1 \leq t_2 \leq \cdots \leq t_k \leq 1. \) Define
\[
 f_{i_1, i_2, \ldots, i_k}(x_1, x_2, \ldots, x_k) := \prod_{s=0}^{s=k-1} \frac{N!}{(i_{s+1} - i_s - 1)!} \frac{(x_{s+1} - x_s)^{i_{s+1} - i_s - 1}}{(i_{s+1} - i_s - 1)!}
\]
with \( x_0 := 0, \) \( x_{k+1} := 1, \) \( i_0 := 0, \) \( i_{k+1} := N + 1. \) Suppose the cumulative distribution function \( F_u(\gamma) := \Pr \{ u(q) \leq \gamma \} \) is continuous. Then
\[
 F(t_1, t_2, \cdots, t_k) = \int_{D_{t_1, t_2, \cdots, t_k}} f_{i_1, i_2, \ldots, i_k}(x_1, x_2, \ldots, x_k) \, dx_1 \, dx_2 \cdots \, dx_k
\]
where
\[ D_{t_1, t_2, \cdots, t_k} := \{(x_1, x_2, \cdots, x_k) : \quad 0 \leq x_1 \leq x_2 \leq \cdots \leq x_k, \ x_s \leq t_s, \ s = 1, 2, \cdots, k}\].

**Remark 4.1** Theorem 4.1 can play an important role in robust control as illustrated in the following sections. However, its further application is limited by the continuity assumption. In many robust control problems, it is reasonable to assume that \( u(q) \) is measurable, while the continuity of \( F_u(\gamma) \) is not necessarily guaranteed. For example, \( F_u(\gamma) \) is not continuous when uncertain quantity \( u(q) \) equals to a constant in an open set of \( Q. \) We can come up with many uncertain systems with which the continuity
assumption for the distribution of quantity \( u(q) \) is not guaranteed. To tackle these problems without continuity assumption by probabilistic approach and investigate the minimum computational effort, we shall develop a distribution inequality which accommodates the case when the continuity is not guaranteed.

First, we shall established the following lemma.

**Lemma 4.1** Let \( U \) be a random variable with uniform distribution over \([0, 1]\) and \( \hat{U}_n, \ n = 1, 2, \ldots, N \) be the order statistics of \( U \), i.e., \( \hat{U}_1 \leq \hat{U}_2 \leq \cdots \leq \hat{U}_N \). Let \( 0 = t_0 < t_1 < t_2 < \cdots < t_k \leq 1 \). Define

\[
G_{j_1, j_2, \ldots, j_k} (t_1, t_2, \ldots, t_k) := (1 - t_k)^{N - \sum_{l=1}^{k} \frac{t_l}{j_l}} \prod_{s=1}^{k} \left( N - \frac{\sum_{l=1}^{s-1} j_l}{j_s} \right) (t_s - t_{s-1})^{j_s}
\]

and

\[
I_{i_1, i_2, \ldots, i_k} := \left\{ (j_1, j_2, \ldots, j_k) : i_s \leq \sum_{l=1}^{s} j_l \leq N, \ s = 1, 2, \ldots, k \right\}.
\]

Then

\[
\Pr \left\{ \hat{U}_{i_1} \leq t_1, \hat{U}_{i_2} \leq t_2, \cdots, \hat{U}_{i_k} \leq t_k \right\} = \sum_{(j_1, j_2, \ldots, j_k) \in I_{i_1, i_2, \ldots, i_k}} G_{j_1, j_2, \ldots, j_k} (t_1, t_2, \ldots, t_k)
\]

**Proof.** Let \( j_s \) be the number of samples of \( U \) which fall into \((t_{s-1}, t_s]\), \( s = 1, 2, 3, \cdots, k \). Then the number of samples of \( U \) which fall into \([0, t_s]\) is \( \sum_{l=1}^{s} j_l \). It is easy to see that the event \( \{ \hat{U}_{i_s} \leq t_s \} \) is equivalent to event \( \{ i_s \leq \sum_{l=1}^{s} j_l \leq N \} \). Furthermore, the event

\[
\left\{ \hat{U}_{i_1} \leq t_1, \hat{U}_{i_2} \leq t_2, \cdots, \hat{U}_{i_k} \leq t_k \right\}
\]
is equivalent to the event \( \{ i_s \leq \sum_{l=1}^{s} j_l \leq N, \ s = 1, 2, \cdots, k \} \). Therefore,

\[
\begin{align*}
\Pr \left\{ \hat{U}_{i_1} \leq t_1, \hat{U}_{i_2} \leq t_2, \cdots, \hat{U}_{i_k} \leq t_k \right\} \\
= \sum_{(j_1, j_2, \cdots, j_k) \in \ell_1, \ell_2, \cdots, \ell_k} \prod_{s=1}^{k} \left( N - \sum_{l=1}^{s-1} j_l \right) (t_s - t_{s-1})^{j_s} (1 - t_k)^{N - \sum_{l=1}^{s} j_l} \\
= \sum_{(j_1, j_2, \cdots, j_k) \in \ell_1, \ell_2, \cdots, \ell_k} G_{j_1, j_2, \cdots, j_k} (t_1, t_2, \cdots, t_k).
\end{align*}
\]

\( \square \)

**Theorem 4.2** Let \( 0 = t_0 \leq t_1 \leq t_2 \leq \cdots \leq t_k \leq 1 \). Define

\[
\tilde{F}(t_1, t_2, \cdots, t_k) := \Pr \{ F_u(\hat{u}_{i_1}) < t_1, \ F_u(\hat{u}_{i_2}) < t_2, \cdots, \ F_u(\hat{u}_{i_k}) < t_k \}
\]

and \( \tau_s := \sup_{x: F_u(x) < t_s} F_u(x), \ s = 1, 2, \cdots, k \). Suppose \( u(q) \) is a measurable function of \( q \). Then

\[
\tilde{F}(t_1, t_2, \cdots, t_k) = \int_{D_{\tau_1, \tau_2, \cdots, \tau_k}} f_{i_1, i_2, \cdots, i_k}(x_1, x_2, \cdots, x_k) \, dx_1 \, dx_2 \cdots \, dx_k.
\]

Furthermore, \( \tilde{F}(t_1, t_2, \cdots, t_k) \leq \int_{D_{t_1, t_2, \cdots, t_k}} f_{i_1, i_2, \cdots, i_k}(x_1, x_2, \cdots, x_k) \, dx_1 \, dx_2 \cdots \, dx_k \) and the equality holds if \( F_u(\gamma) \) is continuous.

**Proof.** Define \( \alpha_0 := -\infty \) and \( \alpha_s := \sup \{ x : F_u(x) < t_s \} \), \( \alpha^-_s := \alpha_s - \epsilon \), \( s = 1, 2, \cdots, k \) where \( \epsilon > 0 \) can be arbitrary small. Let \( \phi_s := F_u(\alpha^-_s), \ s = 1, 2, \cdots, k \). We can show that \( \phi_l < \phi_s \) if \( \alpha_l < \alpha_s \), \( 1 \leq l < s \leq k \). In fact, if this is not true, we have \( \phi_l = \phi_s \). Because \( \epsilon \) can be arbitrary small, we have \( \alpha^-_s \in (\alpha_l, \alpha_s) \). Notice that \( \alpha_l = \min \{ x : F_u(x) \geq t_l \} \), we have \( t_l \leq \phi_s = \phi_l \). On the other hand, by definition we know that \( \alpha^-_l \in \{ x : F_u(x) < t_l \} \) and thus \( \phi_l = F_u(\alpha^-_l) < t_l \), which is a contradiction. Notice that \( F_u(\gamma) \) is nondecreasing and
right-continuous, we have $a_1 \leq a_2 \leq \cdots \leq a_k$ and $0 \leq \phi_1 \leq \phi_2 \leq \cdots \leq \phi_k \leq 1$ and that event $\{F_{u}(\hat{u}_{i_s}) < t_s\}$ is equivalent to the event $\{\hat{u}_{i_s} < a_s\}$. Furthermore, event $\{F_{u}(\hat{u}_{i_1}) < t_1, F_{u}(\hat{u}_{i_2}) < t_2, \cdots, F_{u}(\hat{u}_{i_k}) < t_k\}$ is equivalent to event $\{\hat{u}_{i_1} < \alpha_1, \hat{u}_{i_2} < \alpha_2, \cdots, \hat{u}_{i_k} < \alpha_k\}$ which is defined by $k$ constraints $\hat{u}_{i_s} < a_s, s = 1, 2, \cdots, k$. For every $l < k$, delete constraint $\hat{u}_{i_l} < \alpha_l$ if there exists $s > l$ such that $\alpha_s = \alpha_l$. Let the remained constraints be $\hat{u}_{i_s} < \alpha_s', s = 1, 2, \cdots, k'$ where $\alpha_1' < \alpha_2' < \cdots < \alpha_k'$. Since all constraints deleted are actually redundant, it follows that event $\{\hat{u}_{i_1} < \alpha_1, \hat{u}_{i_2} < \alpha_2, \cdots, \hat{u}_{i_k} < \alpha_k\}$ is equivalent to event $\{\hat{u}_{i_1} < \alpha_1', \hat{u}_{i_2} < \alpha_2', \cdots, \hat{u}_{i_k'} < \alpha_k'\}$. Now let $j_s$ be the number of observations of $u(q)$ which fall into $[\alpha_{s-1}', \alpha_s')$, $s = 1, 2, \cdots, k'$. Then the number of observations of $u(q)$ which fall into $(-\infty, \alpha_s')$ is $\sum_{i=1}^{j_s} j_i$. It is easy to see that the event $\{\hat{u}_{i_s'} < \alpha_s'\}$ is equivalent to the event $\{i_s' \leq \sum_{i=1}^{j_s} j_i \leq N\}$. Furthermore, the event $\{\hat{u}_{i_1} < \alpha_1', \hat{u}_{i_2} < \alpha_2', \cdots, \hat{u}_{i_k'} < \alpha_k'\}$ is equivalent to event

$$\left\{i_s' \leq \sum_{i=1}^{j_s} j_i \leq N, \ s = 1, 2, \cdots, k'\right\}.$$ 

Therefore

$$F(t_1, t_2, \cdots, t_k) = \Pr \{F_{u}(\hat{u}_{i_1}) < t_1, F_{u}(\hat{u}_{i_2}) < t_2, \cdots, F_{u}(\hat{u}_{i_k}) < t_k\}$$

$$= \Pr \{\hat{u}_{i_1} < \alpha_1, \hat{u}_{i_2} < \alpha_2, \cdots, \hat{u}_{i_k} < \alpha_k\}$$

$$= \Pr \{\hat{u}_{i_1'} < \alpha_1', \hat{u}_{i_2'} < \alpha_2', \cdots, \hat{u}_{i_k'} < \alpha_k'\}$$

$$= \sum_{(j_1, j_2, \cdots, j_{k'}) = 1}^{k'} \prod_{s=1}^{k'} \left(N - \sum_{i=1}^{j_s} j_i\right) \left[F_{u}(\alpha_{s'}') - F_{u}(\alpha_{s-1}')\right]^{i_s'}$$

$$\times \left[1 - F_{u}(\alpha_{k'}')\right]^{N - \sum_{i=1}^{k'} j_i}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[
= \sum_{(j_1,j_2,\ldots,j_{k'}) \in I_{i_1',j_2',\ldots,j_{k'}}} G_{j_1,j_2,\ldots,j_{k'}} (\phi'_1, \phi'_2, \ldots, \phi'_{k'}). 
\]

Now consider event \( \{ \hat{U}_{i_1} \leq \phi_1, \hat{U}_{i_2} \leq \phi_2, \ldots, \hat{U}_{i_k} \leq \phi_k \} \). For every \( l < k \), delete constraint \( \hat{U}_{i_l} \leq \phi_l \) if there exists \( s > l \) such that \( \phi_s = \phi_l \). Notice that \( \phi_s = F_u(\alpha_s^-) \) and \( \phi_l < \phi_s \) if \( \alpha_l < \alpha_s \), \( 1 \leq l < s \leq k \), the remained constraints must be \( \hat{U}_{i_s} \leq \phi_s', s = 1, 2, \ldots, k' \) where \( \phi_s = F_u(\alpha_s^-) \), \( s = 1, 2, \ldots, k' \) and \( \phi_1 < \phi_2 < \cdots < \phi_{k'} \). Since all constraints deleted are actually redundant, it follows that event \( \{ \hat{U}_{i_1} \leq \phi_1, \hat{U}_{i_2} \leq \phi_2, \ldots, \hat{U}_{i_k} \leq \phi_k \} \) is equivalent to event
\[
\{ \hat{U}_{i_1} \leq \phi'_1, \hat{U}_{i_2} \leq \phi'_2, \ldots, \hat{U}_{i_{k'}} \leq \phi'_{k'} \}. 
\]

By Theorem 2.2.3 in [20] and Lemma 4.1
\[
\int_{D_{\phi_1,\phi_2,\ldots,\phi_k}} f_{i_1,i_2,\ldots,i_k}(x_1, x_2, \ldots, x_k) \, dx_1 dx_2 \cdots dx_k = \Pr \left\{ \{ \hat{U}_{i_1} \leq \phi_1, \hat{U}_{i_2} \leq \phi_2, \ldots, \hat{U}_{i_k} \leq \phi_k \} \right\}
= \Pr \left\{ \{ \hat{U}_{i_1}' \leq \phi'_1, \hat{U}_{i_2}' \leq \phi'_2, \ldots, \hat{U}_{i_{k'}}' \leq \phi'_{k'} \} \right\}
= \sum_{(j_1,j_2,\ldots,j_{k'}) \in I_{i_1',j_2',\ldots,j_{k'}}} G_{j_1,j_2,\ldots,j_{k'}} (\phi'_1, \phi'_2, \ldots, \phi'_{k'}). 
\]

Therefore, \( \tilde{F}(t_1, t_2, \ldots, t_k) = \int_{D_{\phi_1,\phi_2,\ldots,\phi_k}} f_{i_1,i_2,\ldots,i_k}(x_1, x_2, \ldots, x_k) \, dx_1 dx_2 \cdots dx_k \). By the definitions of \( \tau_s \) and \( \phi_s \), we know that \( D_{\tau_1,\tau_2,\ldots,\tau_k} \) is the closure of \( D_{\phi_1,\phi_2,\ldots,\phi_k} \), i.e., \( D_{\tau_1,\tau_2,\ldots,\tau_k} = \overline{D_{\phi_1,\phi_2,\ldots,\phi_k}} \) and that their Lebesgue measures are equal. It follows that
\[
\tilde{F}(t_1, t_2, \ldots, t_k) = \int_{D_{\tau_1,\tau_2,\ldots,\tau_k}} f_{i_1,i_2,\ldots,i_k}(x_1, x_2, \ldots, x_k) \, dx_1 dx_2 \cdots dx_k.
\]
Notice that $\tau_s \leq t_s, \ s = 1, 2, \ldots, k$, we have $D_{\tau_1, \tau_2, \ldots, \tau_k} \subseteq D_{t_1, t_2, \ldots, t_k}$ and hence

$$\tilde{F}(t_1, t_2, \ldots, t_k) \leq \int_{D_{t_1, t_2, \ldots, t_k}} f_{i_1, i_2, \ldots, i_k}(x_1, x_2, \ldots, x_k) \, dx_1 \, dx_2 \cdots \, dx_k.$$ 

Furthermore, if $F_u(\gamma)$ is continuous, then $\tau_s = t_s, \ s = 1, 2, \ldots, k$, hence $D_{\tau_1, \tau_2, \ldots, \tau_k} = D_{t_1, t_2, \ldots, t_k}$ and the equality holds. \hfill \Box

### 4.3 Performance and Confidence Tradeoff

For the synthesis of a controller for an uncertain system, we usually have a conflict between the performance level and robustness. The following theorem helps to make the tradeoff.

**Theorem 4.3** Let $1 \leq n \leq N, \ 1 \leq m \leq N$ and $\varepsilon \in (0, 1)$. Suppose $u(q)$ is measurable. Then

$$\Pr \{\Pr \{u(q) > \hat{u}_n\} \leq \varepsilon\} \geq 1 - \int_0^{1-\varepsilon} \frac{N!}{(n-1)!(N-n)!} x^{n-1}(1-x)^{N-n} \, dx \quad (4.1)$$

and the equality holds if and only if $\sup_{x: F_u(x) < 1-\varepsilon} F_u(x) = 1 - \varepsilon$; Moreover,

$$\Pr \{\Pr \{u(q) < \hat{u}_m\} \leq \varepsilon\} \geq 1 - \int_0^{1-\varepsilon} \frac{N!}{(m-1)!(N-m)!} x^{m-1}(1-x)^{N-m-1} \, dx \quad (4.2)$$

and the equality holds if and only if $\inf_{x: F_u(x) > \varepsilon} F_u(x) = \varepsilon$.

**Proof.** Apply Theorem 4.2 to the case of $k = 1, \ i_1 = n$, we have

$$\Pr \{F_u(\hat{u}_n) < 1 - \varepsilon\} = \int_{D_r} \frac{N!}{(n-1)!(N-n)!} x^{n-1}(1-x)^{N-n} \, dx$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[
\leq \int_0^{1-\varepsilon} \frac{N!}{(n-1)!(N-n)!} x^{n-1}(1-x)^{N-n} dx
\]

where \(D_\tau = (0, \tau]\) with \(\tau = \sup_{x: F_u(x) < 1-\varepsilon} F_u(x)\). Therefore,

\[
\Pr \{ \Pr \{ u(q) > \hat{u}_n \} \leq \varepsilon \} = \Pr \{ F_u(\hat{u}_n) \geq 1 - \varepsilon \}
= 1 - \Pr \{ F_u(\hat{u}_n) < 1 - \varepsilon \} \geq 1 - \int_0^{1-\varepsilon} \frac{N!}{(n-1)!(N-n)!} x^{n-1}(1-x)^{N-n} dx.
\]

The equality holds if and only if \(\sup_{x: F_u(x) < 1-\varepsilon} F_u(x) = 1 - \varepsilon\) because \(D_\tau = (0, 1-\varepsilon]\)

if and only if \(\tau = \sup_{x: F_u(x) < 1-\varepsilon} F_u(x) = 1 - \varepsilon\).

Now let \(v(q) = -u(q)\). Let the cumulative distribution function of \(v(q)\) be \(F_v(.)\) and define order statistics \(\hat{v}_i, \; i = 1, 2 \cdots, N\) as the \(i\)-th smallest observation of \(v(q)\) during \(N\) i.i.d. sample experiments, i.e., \(\hat{v}_1 \leq \hat{v}_2 \leq \hat{v}_3 \leq \cdots \leq \hat{v}_N\). Obviously, \(\hat{u}_m = -\hat{v}_{N+1-m}\) for any \(1 \leq m \leq N\). It is also clear that \(F_v(-x) = 1 - F_u(x^-)\), which leads to the equivalence of \(\sup_{x: F_u(x) < 1-\varepsilon} F_v(x) = 1 - \varepsilon\) and \(\inf_{x: F_u(x) > \varepsilon} F_u(x) = \varepsilon\). Therefore, apply (4.1) to the situation of \(v(q)\), we have

\[
\Pr \{ \Pr \{ u(q) < \hat{u}_m \} \leq \varepsilon \}
= \Pr \{ \Pr \{ v(q) > \hat{v}_{N+1-m} \} \leq \varepsilon \} \geq 1 - \int_0^{1-\varepsilon} \frac{N!}{(m-1)!(N-m)!} x^{N-m}(1-x)^{m-1} dx
\]

and the equality holds if and only if \(\inf_{x: F_u(x) > \varepsilon} F_u(x) = \varepsilon\).

In Figure 4.3 we computed the lower bound for \(\Pr \{ u(q) > \hat{u}_n \} \leq \varepsilon\) for sample size \(N = 8000\) with \(\varepsilon = 0.0010\), \(\varepsilon = 0.0012\), and \(\varepsilon = 0.0015\) respectively. We can see that the performance level increases as \(n\) decreases, while confidence level decreases.
Theorem 4.3 can be used as a guideline to robust control analysis and synthesis. For example, when dealing with robust stability problem, we need to compute the maximum of the real part of the closed-loop poles, denoted by $u(q)$, which is a function of uncertain parameters $q$. When we estimate the upper bound of $u(q)$ by sampling, it is possible that most of the samples concentrated in an interval and very few fall far beyond that interval. If we take $\hat{u}_N$ as the maximum in the design of a controller, it may be conservative. However, if we choose $n$ such that $\hat{u}_n$ is much smaller than $\hat{u}_N$, while $\Pr \{ \Pr \{ u(q) > \hat{u}_n \} \leq \varepsilon \}$ is close to 1, then the controller based on $\hat{u}_n$ may have much better performance but with only a little bit more increase of risk. For example, let's say, sample size $N = 8000$ and $\varepsilon = 0.0010$, the distribution of sample is like this, $\hat{u}_1$, $\hat{u}_2$, $\ldots$, $\hat{u}_{7900}$ concentrated in an interval, and $\hat{u}_{7900}$ is much smaller than $\hat{u}_{7901}$, $\hat{u}_{7902}$, $\ldots$, $\hat{u}_{8000}$. It is sure that the controller designed by taking $\hat{u}_{7900}$ as the upper bound will have much higher performance level than the controller designed by taking $\hat{u}_{8000}$ as the upper bound. To compare
the risks for these two cases, we have \( \Pr \{ \Pr \{ u(q) > \hat{u}_{8000} \} \leq 0.0010 \} \geq 0.99966 \) and \( \Pr \{ \Pr \{ u(q) > \hat{u}_{7900} \} \leq 0.0010 \} \geq 0.99951 \). These data indicate that there is only a little bit more increase of risk by taking \( \hat{u}_{7900} \) instead of \( \hat{u}_{8000} \) as the upper bound in designing a controller.

4.4 **MINIMUM SAMPLE SIZE**

In addition to the situation of making the tradeoff between performance degradation and risk, Theorem 4.3 can also play an important role in the issue of computational effort required to come up with an estimate of the upper bound (or lower bound) of a quantity with a certain accuracy and confidence. This issue was first addressed independently by Tempo, Bai and Dabbene in [45] and Khargonekar and Tikku in [33] and their results are summarized in the following theorem.

**Theorem 4.4** For any \( \varepsilon, \delta \in (0, 1) \), \( \Pr \{ \Pr \{ u(q) > \hat{u}_N \} \leq \varepsilon \} \geq 1 - \delta \) if \( N \geq \frac{\ln \frac{1}{\delta}}{\ln \frac{1}{1 - \varepsilon}} \).

**Remark 4.2** This theorem only answers the question that how much computational effort is sufficient. We shall also be concerned about what is the minimum computational effort. By applying Theorem 4.3 to the case of \( n = N \), we can also obtain Theorem 4.4. Moreover, we can see that, for a certain accuracy (i.e., a fixed value of \( \varepsilon \)), the bound becomes minimum if and only if

\[
\sup_{\{x: F_u(x) < 1 - \varepsilon\}} F_u(x) = 1 - \varepsilon. \tag{4.3}
\]

If \( F_u(.) \) is continuous (i.e., (4.3) is guaranteed for any \( \varepsilon \in (0, 1) \)), the bound is of course tight.
Similarly, by applying Theorem 4.3 to the case of \( m = 1 \) we have that for \( \varepsilon, \delta \in (0,1) \),

\[
\Pr \{ \Pr \{ \hat{u}(q) \geq \hat{u}_1 \} \geq 1 - \varepsilon \} \geq 1 - \delta
\]

if \( N \geq \frac{\ln \frac{1}{\delta}}{\ln \frac{1}{1 - \varepsilon}} \). For a fixed \( \varepsilon \in (0,1) \), this bound is tight if and only if \( \inf \{ x : F_u(x) > \varepsilon \} F_u(x) = \varepsilon \).

### 4.5 Quantity Range and Distribution-Free Tolerance Intervals

To estimate the range of an uncertain quantity with a certain accuracy and confidence level apriori specified, we have the following corollary.

**Corollary 4.1** Suppose \( F_u(\gamma) \) is continuous. For any \( \varepsilon, \delta \in (0,1) \),

\[
\Pr \{ \Pr \{ \hat{u}_1 < u(q) \leq \hat{u}_N \} \geq 1 - \varepsilon \} \geq 1 - \delta
\]

if and only if \( \mu(N) \leq \delta \) where \( \mu(N) := (1 - \varepsilon)^{N-1} \left[ 1 + (N - 1)\varepsilon \right] \) is a monotonically decreasing function of \( N \).

**Remark 4.3** The minimum \( N \) guaranteeing this condition can be found by a simple bisection search. This bound of sample size is minimum because our computation of probability is exact. This bound is also practically small, for example, \( N \geq 1,483 \) if \( \varepsilon = \delta = 0.005 \), and \( N \geq 9,230 \) if \( \varepsilon = \delta = 0.001 \). Therefore, to obtain a reliable estimate of the range of an uncertain quantity, computational complexity is not an issue.

In general, it is important to know the probability of a quantity falling between two arbitrary samples. To that end, we have
Corollary 4.2 \textit{Suppose }F_u(\gamma)\textit{ is continuous. Then for }\varepsilon \in (0, 1)\textit{ and }1 \leq m < n \leq N,\textit{ }

\Pr\{\Pr\{\hat{u}_m < u(q) \leq \hat{u}_n\} \geq 1 - \varepsilon\} = \int_{1-\varepsilon}^{1} N\left(\frac{N - 1}{n - m - 1}\right)x^{n-m-1}(1 - x)^{N-n+m}dx.\textit{ }

Here } (\hat{u}_m, \hat{u}_n) \textit{ is referred as distribution-free tolerance interval in order statistics theory (see [20]).}
CHAPTER 5

CONSTRAINED OPTIMAL SYNTHESIS AND ROBUSTNESS ANALYSIS

In the last chapter, we have considered the robust analysis and synthesis problems without constraints. In general, most robust control problems are associated with complex constraints. Henceforth, we shall address this issue in this chapter.

In this chapter, we consider robust control using randomized algorithms. We extend the existing order statistics distribution theory to the general case in which the distribution of population is not assumed to be continuous and the order statistics is associated with certain constraints. In particular, we derive an inequality on distribution for related order statistics. Moreover, we also propose two different approaches in searching reliable solutions to the robust analysis and optimal synthesis problems under constraints. Furthermore, minimum computational effort is investigated and bounds for sample size are derived.

The robust control analysis and synthesis problems under constraints are, in general, very hard to deal with in the deterministic framework. For example, it is well-known that a multi-objective control problem involving mixed $H_2$ and $H_\infty$ objectives are very hard to solve even though there are elegant solutions to the pure $H_2$ or $H_\infty$ problems [53]. In addition to its low computational complexity, the advantages of randomized algorithms can be found in the flexibility and adaptiveness in dealing with control analysis or synthesis problems with complicated constraints or in the situation of handling nonlinearities.
In this chapter, we first show that most of the robust control problems can be formulated as constrained optimal synthesis or robust analysis problems. Since the exact robust analysis or synthesis is, in general, impossible, we seek a ‘reliable’ solution by using randomized algorithms. Roughly speaking, by ‘reliability’ we mean how the solution resulted by randomized algorithms approaches the exact one. In this chapter, we measure the degree of ‘reliability’ in terms of accuracy $1 - \varepsilon$ and confidence level $1 - \delta$. Actually, terminologies like ‘accuracy’ and ‘confidence level’ have been used in [45] and [33] where accuracy $1 - \varepsilon$ is referred as an upper bound of the absolute volume of a subset of parameter space $\mathbf{Q}$. However, in this chapter, we emphasis that the accuracy $1 - \varepsilon$ is an upper bound for the ratio of volume of the constrained subset $\mathbf{Q}_C := \{ \text{constraint set \mathbf{C} holds, } q \in \mathbf{Q}\}$ with respect to the volume of parameter space $\mathbf{Q}$. For example, when estimating the minimum of a quantity $u(q)$ over $\mathbf{Q}_C$, the ratio may be $\frac{\text{volume of } \{u(q) \geq \hat{u}_\text{min}, \ q \in \mathbf{Q}_C\}}{\text{volume of } \mathbf{Q}_C}$ where $\hat{u}_\text{min}$ is an estimate resulted by randomized algorithms for quantity $u(q)$. We can see that the ratio of volume is a better indicator of the ‘reliability’ than the absolute volume of $\{u(q) \geq \hat{u}_\text{min}, \ q \in \mathbf{Q}_C\}$.

Based on this measure of ‘reliability’, we propose two different approaches aimed at seeking a solution to the robust analysis or optimal synthesis problem with a certain a priori specified degree of ‘reliability’. One is the direct approach. The key issue is to determine the number of samples needed to be generated from the parameter space $\mathbf{Q}$ for a given reliability measure. Actually, Khargonekar and Tikku in [33] have applied similar approach to stability margin problem, though the measure of ‘reliability’ is in terms of the absolute volume. In that paper, a sufficient condition is derived on the sample size required to come up with a ‘reliable’ estimate of the robust stability margin (See Theorem 3.3 in [33]). In this chapter, we also
derive the bound of sample size and give the sufficient and necessary condition for the existence of minimum distribution-free samples size. Our result shows that, the bound of sample size necessarily involves $\rho := \text{volume of } Q_C$. Thus estimating $\rho$ becomes essential. Unfortunately, estimating $\rho$ is time-consuming and the resulted sample size is not accurate. To overcome this difficulty, we propose and strongly advocate another approach—the indirect approach. The key issue is to determine the constrained sample size, which is the number of samples needed that fall into the constrained subset $Q_C$. We derive bounds of constrained sample size and give the sufficient and necessary condition for the existence of minimum distribution-free constrained samples size. The bounds do not involve $\rho$ and can be computed exactly. This result makes it possible to obtain a reliable solution without estimating the volume of the constrained parameter subset $Q_C$.

This chapter is organized as follows. Section 5.1 presents the problem formulation and motivations. In Section 5.2, we derive the exact distribution of related order statistics without the continuity assumption. Distribution free tolerance interval and estimation of quantity range is discussed in Section 5.3. Section 5.4 gives the minimum sample size under various assumptions.

5.1 Preliminary and Problem Formulation

Let $q = [q_1 \cdots q_n]^T$ be a vector of a control system's parameters, bounded in a compact set $Q$, i.e., $q \in Q$. Let $C$ be a set of constraints that $q$ must satisfy. Define the constrained subset of $Q$ by $Q_C := \{ C \text{ holds, } q \in Q \}$. Let $u(q)$ denote a performance index function. In many applications, we are concerned with a performance index function $u(q)$ of the system under the set of constraints $C$. It is natural to ask the following questions:
What is \( \min_{Q_C} u(q) \) (or \( \max_{Q_C} u(q) \))?

What is the value of \( q \) at which \( u(q) \) achieves \( \min_{Q_C} u(q) \) (or \( \max_{Q_C} u(q) \))?

Consider, for example, an uncertain system shown in Figure 4.1. Denote the transfer function from \( v \) to \( z \) by \( T_{zv} \) and suppose that \( T_{zv} \) has the following state space realization

\[
T_{zv} = \begin{bmatrix}
    A(q) & B(q) \\
    C(q) & D(q)
\end{bmatrix}.
\]

We can now consider several robustness problems:

- Robust stability: Let \( Q_C = Q \) and \( u(q) := \max_i \Re \lambda_i(A(q)) \) where \( \lambda_i(A) \) denotes the \( i \)-th eigenvalue of \( A \). Then the system is robustly stable if

\[
\max_{q \in Q_C} u(q) < 0.
\]

- Stability margin: Assume that \( \Delta(q) \) belongs to the class of allowable perturbations \( \Delta \) which has a certain block structure. For a given real number \( \gamma \), let \( \Delta_\gamma \) denote the subset of perturbations in \( \Delta \) with size at most \( \gamma \), i.e.,

\[
\Delta_\gamma := \{ \Delta(q) \in \Delta : \bar{\sigma}(\Delta(q)) \leq \gamma \}.
\]

The robustness measure \( \gamma_{opt} \) is defined as the smallest allowable perturbation that destabilizes the feedback interconnection. Let \( \gamma_0 \) be an upper bound for \( \gamma_{opt} \). Define parameter space \( Q \) by

\[
Q := \{ q : \Delta(q) \in \Delta_\gamma \}
\]

and constrained subset \( Q_C \) by

\[
Q_C := \{ q : q \in Q \text{ and } A(q) \text{ is unstable} \}.
\]

Let \( u(q) := \bar{\sigma}(\Delta(q)) \). It follows that the stability margin problem is equivalent to computing \( \gamma_{opt} = \min_{Q_C} u(q) \).
• Robust performance: Suppose $A(q)$ is stable for all $q \in Q$. Define $u(q) := ||T_{zv}||_\infty$. Then the robust performance problem is to determine if $\max_{q \in Q} u(q) \leq \gamma$ is satisfied for some prespecified $\gamma > 0$.

• Performance range: Let $Q_C \subseteq Q$ be a given set of parameters such that $A(q)$ is stable for all $q \in Q_C$. Define again $u(q) := ||T_{zv}||_\infty$. Then the problem of determining the range of the system's $H_\infty$ performance level can be formulated as finding $\min_{q \in Q_C} u(q)$ and $\max_{q \in Q_C} u(q)$.

As another example, consider the problem of designing a controller $K(q)$ for an uncertain system $P(s)$. Suppose that $q$ is a vector of controller parameters to be designed and that the controller is connected with $P(s)$ in a lower LFT setup. Let the transfer function of the whole system be denoted as $F_l(P(s), K(q))$. Suppose that $F_l(P(s), K(q))$ has the following state space realization

$$F_l(P(s), K(q)) = \begin{bmatrix} A_s(q) & B_s(q) \\ C_s(q) & D_s(q) \end{bmatrix}.$$

Then we can formulate the problem as a constrained optimal synthesis problem by defining a performance index $u(q) := ||F_l(P(s), K(q))||_\infty$ and restricting parameter $q$ to $Q_C := \{ \max \Re \lambda_i(A_s(q)) < -\alpha, \, q \in Q \}$ where $\alpha > 0$ is not too small for a stability margin. Then the $H_\infty$ design problem is to determine a vector of parameters achieving $\min_{q \in Q_C} u(q)$.

5.1.1 A Measure of Reliability

Since the exact solution to the analysis or synthesis problem is impossible. Measuring how the solution resulted by the randomized algorithm approaches the exact one
becomes essential. We shall first introduce the concept of \textit{volume} proposed in [33]. Let $w(q)$ denote the cumulative distribution function of $q$. For a subset $U \subseteq Q$, the volume of $U$, denoted by $vol_w\{U\}$, is defined by $vol_w\{U\} := \int_{q \in U} dw(q)$. Define $\rho := \frac{vol_w\{Q_C\}}{vol_w\{Q\}}$. Then it follows that $\rho = vol_w\{Q_C\}$ since $vol_w\{Q\} = 1$. We assume throughout this chapter that $u(q)$ is a measurable function of $q$ and that $vol_w\{Q_C\} > 0$. We also assume throughout this chapter that $\varepsilon, \delta \in (0, 1)$. Let $\hat{u}_{\text{min}}$ and $\hat{u}_{\text{max}}$ be the estimates of $\min_{q \in Q_C} u(q)$ and $\max_{q \in Q_C} u(q)$ respectively. Note that $\hat{u}_{\text{max}}$ and $\hat{u}_{\text{min}}$ are random variables resulted by randomized algorithms. A reliable estimate of $\hat{u}_{\text{min}}$ should guarantee $\Pr\{\frac{vol_w\{u(q) \geq \hat{u}_{\text{min}}, q \in Q_C\}}{vol_w\{Q_C\}} \geq 1 - \varepsilon\} \geq 1 - \delta$ for a small $\varepsilon$ and a small $\delta$. Similarly, a reliable estimate of $\hat{u}_{\text{max}}$ should guarantee $\Pr\{\frac{vol_w\{u(q) \leq \hat{u}_{\text{max}}, q \in Q_C\}}{vol_w\{Q_C\}} \geq 1 - \varepsilon\} \geq 1 - \delta$ for a small $\varepsilon$ and a small $\delta$.

5.1.2 Two Different Approaches

- **Indirect Approach** Generate i.i.d. samples $q^i$ for $q$ by the same distribution function $w(q)$. Continue the sampling process until we obtain $N_c$ observations of $q$ which belong to $Q_C$. Let $L$ be the number of i.i.d. experiments when this sampling process is terminated. Then $L$ is a random number with distribution satisfying $\sum_{l=N_c}^{\infty} \Pr\{L = l\} = 1$ and we can show that $E[L] = \frac{N_c}{\rho}$. Let the observations which belong to $Q_C$ be denoted as $q_C^i, \ i = 1, \cdots, N_c$. Define order statistics $\hat{u}_i, \ i = 1, \cdots, N_c$ as the $i$th smallest one of the set of observations $\{u(q_C^i) : \ i = 1, \cdots, N_c\}$, i.e., $\hat{u}_1 \leq \cdots \leq \hat{u}_{N_c}$. Obviously, it is reasonable to take $\hat{u}_1$ as an estimate for $\min_{Q_C} u(q)$ and $\hat{u}_{N_c}$ as an estimate for $\max_{Q_C} u(q)$ if $N_c$ is sufficiently large. Henceforth, we need to know $N_c$ which guarantees $\Pr\{\frac{vol_w\{u(q) \geq \hat{u}_1, q \in Q_C\}}{vol_w\{Q_C\}} \geq 1 - \varepsilon\} \geq 1 - \delta$ and (or) $\Pr\{\frac{vol_w\{u(q) \leq \hat{u}_{N_c}, q \in Q_C\}}{vol_w\{Q_C\}} \geq 1 - \varepsilon\} \geq 1 - \delta$. We call $N_c$ \textit{constrained sample size}. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
- **Direct Approach** Let \( q^1, \cdots, q^N \) be \( N \) i.i.d samples generated by the same distribution function \( w(q) \). Define \( S := \{q^1, \cdots, q^N\} \cap Q_C \). Let \( M \) be the number of the elements in \( S \). Then \( M \) is a random number. If \( M \geq 1 \) we denote the elements of \( S \) as \( q_i^i, \ i = 1, \cdots, M \). Define order statistics \( \hat{u}_i, \ i = 1, \cdots, M \) as the \( i \)th smallest one of the set of observations \( \{u(q_i^i) : i = 1, \cdots, M\} \), i.e., \( \hat{u}_1 \leq \cdots \leq \hat{u}_M \). In particular, let \( \hat{u}_{\text{min}} = \hat{u}_1 \) and \( \hat{u}_{\text{max}} = \hat{u}_M \). We need to know \( N \) which guarantees \( \Pr \left\{ \frac{\text{vol}_w\{u(q) \geq \hat{u}_{\text{min}}, q \in Q_C\}}{\text{vol}_w(Q_C)} \geq 1 - \varepsilon \right\} \geq 1 - \delta \) and (or) \( \Pr \left\{ \frac{\text{vol}_w\{u(q) \leq \hat{u}_{\text{max}}, q \in Q_C\}}{\text{vol}_w(Q_C)} \geq 1 - \varepsilon \right\} \geq 1 - \delta \). We call \( N \) global sample size.

### 5.2 Exact Distribution

Define \( F_u(\gamma) := \frac{\text{vol}_w\{u(q) \leq \gamma, q \in Q_C\}}{\text{vol}_w(Q_C)} \). To compute the probabilities involved in Section 5.1, it is important to know the associated distribution of any \( k \) random variables \( F_u(\hat{u}_{i_1}), \cdots, F_u(\hat{u}_{i_k}), \ 1 \leq i_1 < \cdots < i_k \leq N_C, \ 1 \leq k \leq N_C \) where \( \hat{u}_{i_s}, s = 1, \cdots, k \) is order statistics in the context of the indirect approach.

**Theorem 5.1** Let \( 0 = t_0 < t_1 \leq \cdots \leq t_k \leq 1 \) and \( x_0 = 0, \ x_{k+1} = 1, \ i_0 = 0, \ i_{k+1} = N + 1 \). Define

\[
f_{i_1, \cdots, i_k}(x_1, \cdots, x_k) := \prod_{s=0}^{i_k} \frac{N!}{(i_{s+1} - i_s - 1)!} \frac{(x_{s+1} - x_s)^{i_{s+1} - i_s - 1}}{x_{s+1} - i_s - 1!}
\]

and

\[
D_{p_1, \cdots, p_k} := \{(x_1, \cdots, x_k) : 0 \leq x_1 \leq \cdots \leq x_k, \ x_s \leq p_s, \ s = 1, \cdots, k\}.
\]

Define

\[
F(t_1, \cdots, t_k) := \Pr \{F_u(\hat{u}_{i_1}) < t_1, \cdots, F_u(\hat{u}_{i_k}) < t_k\}
\]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
and \( \tau_s := \sup_{x : F_u(x) < t_s} F_u(x) \), \( s = 1, \cdots, k \). Then

\[
F(t_1, \cdots, t_k) = \int_{D_{t_1, \cdots, t_k}} f_{i_1, \cdots, i_k}(x_1, \cdots, x_k) \, dx_1 \cdots dx_k \leq \int_{D_{t_1, \cdots, t_k}} f_{i_1, \cdots, i_k}(x_1, \cdots, x_k) \, dx_1 \cdots dx_k
\]

and the last equality holds if and only if

\( \exists x_s^* \) such that \( \Pr\{u(q) < x_s^* \mid q \in Q_C\} = t_s \), \( s = 1, \cdots, k \).

**Proof.** Define \( \alpha_0 := -\infty \) and \( \alpha_s := \sup \{x : F_u(x) < t_s\} \), \( \alpha_s^- := \alpha_s - \epsilon \), \( s = 1, \cdots, k \) where \( \epsilon > 0 \) can be arbitrarily small. Let \( \phi_s := F_u(\alpha_s^-) \), \( s = 1, \cdots, k \). We can show that \( \phi_l < \phi_s \) if \( \alpha_l < \alpha_s \), \( 1 \leq l < s \leq k \). In fact, if this is not true, we have \( \phi_l = \phi_s \). Because \( \epsilon \) can be arbitrarily small, we have \( \alpha_s^- \in (\alpha_l, \alpha_s) \). Notice that \( \alpha_l = \min \{x : F_u(x) \geq t_l\} \), we have \( t_l \leq \phi_s = \phi_l \). On the other hand, by definition we know that \( \alpha_s^- \in \{x : F_u(x) < t_l\} \) and thus \( \phi_l = F_u(\alpha_s^-) < t_l \), which is a contradiction. Notice that \( F_u(\gamma) \) is nondecreasing and right-continuous, we have \( \alpha_1 \leq \cdots \leq \alpha_k \) and \( 0 \leq \phi_1 \leq \cdots \leq \phi_k \leq 1 \) and that event \( \{F_u(\hat{u}_{i_s}) < t_s \mid L = l\} \) is equivalent to the event \( \{\hat{u}_{i_s} < \alpha_s \mid L = l\} \). Furthermore, event

\[
\{F_u(\hat{u}_{i_1}) < t_1, \cdots, F_u(\hat{u}_{i_k}) < t_k \mid L = l\}
\]

is equivalent to event

\[
\{\hat{u}_{i_1} < \alpha_1, \cdots, \hat{u}_{i_k} < \alpha_k \mid L = l\}
\]
which is defined by \( k \) constraints \( \hat{u}_{is} < \alpha_s, \ s = 1, \cdots, k \). For every \( l < k \), delete constraint \( \hat{u}_{il} < \alpha_l \) if there exists \( s > l \) such that \( \alpha_s = \alpha_l \). Let the remaining constraints be \( \hat{u}_{is} < \alpha'_s, \ s = 1, \cdots, k' \) where \( \alpha'_1 < \cdots < \alpha'_{k'} \). Since all constraints deleted are actually redundant, it follows that event

\[
\{ \hat{u}_{i_1} < \alpha_1, \cdots, \hat{u}_{i_k} < \alpha_k | L = l \}
\]

is equivalent to event

\[
\{ \hat{u}_{i'_1} < \alpha'_{i'_1}, \cdots, \hat{u}_{i'_{k'}} < \alpha'_{k'} | L = l \}. \]

Now let \( j_s \) be the number of observations \( u(q^s) \) which fall into \([\alpha'_{s-1}, \alpha'_s)\), \( s = 1, \cdots, k' \). Then the number of observations \( u(q^s) \) which fall into \((-\infty, \alpha'_s)\) is \( \sum_{l=1}^s j_l \). It is easy to see that the event

\[
\{ \hat{u}_{i'_s} < \alpha'_s | L = l \}
\]

is equivalent to the event

\[
\left\{ i'_s \leq \sum_{l=1}^s j_l \leq N \right\}.
\]

Furthermore, the event

\[
\{ \hat{u}_{i'_1} < \alpha'_1, \cdots, \hat{u}_{i'_{k'}} < \alpha'_{k'} | L = l \}
\]

is equivalent to event

\[
\left\{ i'_s \leq \sum_{l=1}^s j_l \leq N, \ s = 1, \cdots, k' \right\}.
\]
Therefore

\[ \Pr \{ F_u(\hat{u}_{i_1}) < t_1, \ldots, F_u(\hat{u}_{i_k}) < t_k \mid L = l \} = \Pr \{ \hat{u}_{i_1} < \alpha_1', \ldots, \hat{u}_{i_k} < \alpha_k' \mid L = l \} \]

\[ = \sum_{(j_1, \ldots, j_k') \in I_{i_1}', \ldots, I_{i_k}'} \prod_{s=1}^{k'} \left( N - \sum_{l=1}^{s-1} j_l \right) \left[ F_u(\alpha_s') - F_u(\alpha_{s-1}') \right]^{j_s} \left[ 1 - F_u(\alpha_k') \right]^{N - \sum_{l=1}^{k'} j_l} \]

Now consider event \( \{ \hat{U}_{i_1} \leq \phi_1, \ldots, \hat{U}_{i_k} \leq \phi_k \} \). For every \( l < k \), delete constraint \( \hat{U}_{i_l} \leq \phi_l \) if there exists \( s > l \) such that \( \phi_s = \phi_l \). Notice that \( \phi_s = F_u(\alpha_s^-) \) and \( \phi_l < \phi_s \) if \( \alpha_l < \alpha_s, 1 \leq l < s \leq k \), the remaining constraints must be \( \hat{U}_{i_s} \leq \phi_s, s = 1, \ldots, k' \) where \( \phi_s = F_u(\alpha_s^-), s = 1, \ldots, k' \) and \( \phi_1 < \cdots < \phi_k' \). Since all constraints deleted are actually redundant, it follows that event \( \{ \hat{U}_{i_1} \leq \phi_1, \cdots, \hat{U}_{i_k} \leq \phi_k \} \) is equivalent to event \( \{ \hat{U}_{i_1} \leq \phi_1', \cdots, \hat{U}_{i_k'} \leq \phi_k' \} \). By Theorem 2.2.3 in [20] and Lemma 4.1

\[ \int_{D_{\phi_1, \ldots, \phi_k}} f_{x_1, \ldots, x_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k = \Pr \{ \hat{U}_{i_1} \leq \phi_1, \ldots, \hat{U}_{i_k} \leq \phi_k \} \]

\[ = \Pr \{ \hat{U}_{i_1}' \leq \phi_1', \ldots, \hat{U}_{i_k}' \leq \phi_k' \} = \sum_{(j_1, \ldots, j_k') \in I_{i_1}', \ldots, I_{i_k}'} G_{j_1, \ldots, j_k'} (\phi_1', \ldots, \phi_k') . \]

Therefore,

\[ \Pr \{ F_u(\hat{u}_{i_1}) < t_1, \ldots, F_u(\hat{u}_{i_k}) < t_k \mid L = l \} = \int_{D_{\phi_1, \ldots, \phi_k}} f_{x_1, \ldots, x_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k . \]

It follows that

\[ F(t_1, \ldots, t_k) = \Pr \{ F_u(\hat{u}_{i_1}) < t_1, \ldots, F_u(\hat{u}_{i_k}) < t_k \} \]
\[ = \sum_{l=0}^{\infty} \Pr\{F_u(\hat{u}_{i_l}) < t_1, \ldots, F_u(\hat{u}_{i_k}) < t_k \mid L = l\} \Pr\{L = l\} \]
\[ = \sum_{l=0}^{\infty} \int_{D_{\phi_1, \ldots, \phi_k}} f_{i_1, \ldots, i_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k \Pr\{L = l\} \]

Notice that \(\sum_{l=0}^{\infty} \Pr\{L = l\} = 1\). We have
\[
F(t_1, \ldots, t_k) = \int_{D_{\tau_1, \ldots, \tau_k}} f_{i_1, \ldots, i_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k \sum_{l=0}^{\infty} \Pr\{L = l\}
\]
\[= \int_{D_{\tau_1, \ldots, \tau_k}} f_{i_1, \ldots, i_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k.\]

By the definitions of \(\tau_s\) and \(\phi_s\), we know that \(D_{\tau_1, \ldots, \tau_k}\) is the closure of \(D_{\phi_1, \phi_2, \ldots, \phi_k}\), i.e., \(D_{\tau_1, \ldots, \tau_k} = \bar{D}_{\phi_1, \phi_2, \ldots, \phi_k}\) and that their Lebesgue measures are equal. It follows that
\[
F(t_1, \ldots, t_k) = \int_{D_{\tau_1, \ldots, \tau_k}} f_{i_1, \ldots, i_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k.
\]

Notice that \(\tau_s \leq t_s, \ s = 1, \ldots, k\), we have \(D_{\tau_1, \ldots, \tau_k} \subseteq D_{t_1, \ldots, t_k}\) and hence
\[
F(t_1, \ldots, t_k) \leq \int_{D_{t_1, t_2, \ldots, t_k}} f_{i_1, \ldots, i_k}(x_1, \ldots, x_k) \, dx_1 \cdots dx_k,
\]
where the equality holds if and only if \(\tau_s = t_s, \ s = 1, \ldots, k\), i.e.,
\[
\exists x^*_s \text{ such that } \Pr\{u(q) < x^*_s \mid q \in Q_C\} = t_s, \ s = 1, \ldots, k.
\]

\[ \square \]

**Remark 5.1** For the special case of \(Q_C = Q\) and that \(F_u(.)\) is absolutely continuous, \(F(t_1, \cdots, t_k)\) can be obtained by combining Probability Integral Transformation Theorem and Theorem 2.2.3 in [20]. However, in robust control problem, the con-
tainty of $F_u(.)$ is not necessarily guaranteed. For example, $F_u(.)$ is not continuous when uncertain quantity $u(q)$ equals to a constant in an open set of $Q_C$. We can come up with many uncertain systems in which the continuity assumption for the distribution of quantity $u(q)$ is not guaranteed. Since it is reasonable to assume that $u(q)$ is measurable, Theorem 5.1 can be applied in general to tackle these problems without continuity assumption by a probabilistic approach. In addition, Theorem 5.1 can be applied to investigate the minimum computational effort to come up with a solution with a certain degree of 'reliability' for robust analysis or optimal synthesis problems under constraints.

From the proof of Theorem 5.1, we can see that $F(t_1, \cdots, t_k)$ is not related to the knowledge of $L$, thus we have the following corollary.

**Corollary 5.1** Let $N_2 \geq N_1 \geq N_c$. Then

$$
\Pr \{F_u(\hat{u}_{i_1}) < t_1, \cdots, F_u(\hat{u}_{i_k}) < t_k \mid N_1 \leq L \leq N_2\} = F(t_1, \cdots, t_k).
$$

### 5.3 Quantity Range and Distribution-Free Tolerance Intervals

In robust analysis or synthesis, it is desirable to know function $F_u(.)$ because it is actually the distribution function of quantity $u(q)$ for $q \in Q_C$. However, the exact computation of function $F_u(.)$ is in general impossible. We shall extract as much as possible the information of $F_u(.)$ from observations $u(q^i_C)$, $i = 1, \cdots, N_c$.

**Theorem 5.2** Define

$$
\mathcal{V}(N_c, i, \varepsilon) := \int_\varepsilon^1 \frac{N_c!}{(i-1)!(N_c-i)!} x^{i-1}(1-x)^{N_c-i} dx
$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
for $1 \leq i \leq N_c$. Then

$$\Pr \left\{ \frac{\text{vol}_w \{ u(q) \geq \hat{u}_m, \ q \in \mathcal{Q}_C \}}{\text{vol}_w \{ \mathcal{Q}_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \mathcal{V}(N_c, m, \varepsilon)$$

with the equality holds if and only if $\exists x^*$ such that $F_u(x^*) = \varepsilon$. Moreover,

$$\Pr \left\{ \frac{\text{vol}_w \{ u(q) \leq \hat{u}_m, \ q \in \mathcal{Q}_C \}}{\text{vol}_w \{ \mathcal{Q}_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \mathcal{V}(N_c, N_c + 1 - m, \varepsilon)$$

with the equality holds if and only if $\exists x^*$ such that $\Pr \{ u(q) < x^* \mid q \in \mathcal{Q}_C \} = 1 - \varepsilon$.

**Proof.** Let $v(q) = -u(q)$. Let the cumulative distribution function of $v(q)$ be $F_v(.)$ and define order statistics $\hat{v}_i, \ i = 1, \cdots, N_c$ as the $i$-th smallest one of the set of observations $\{ v(q^i) \mid i = 1, \cdots, N_c \}$, i.e., $\hat{v}_1 \leq \cdots \leq \hat{v}_{N_c}$. Obviously, $\hat{v}_m = -\hat{v}_{N_c+1-m}$ for any $1 \leq m \leq N_c$. It is also clear that $F_v(-x) = 1 - F_u(x^-)$, which leads to $\sup_{\{x:F_v(x)<1-\varepsilon\}} F_v(x) = 1 - \varepsilon \iff \inf_{\{x:F_u(x)\geq\varepsilon\}} F_u(x) = \varepsilon$. Apply Theorem 5.1 to the case of $k = 1, m = N_c + 1 - m$, we have

$$\Pr \{ F_v(\hat{v}_{N_c+1-m}) < 1 - \varepsilon \} = \int_0^\tau \frac{N_c!}{(N_c - m)!(m-1)!} x^{N_c-m}(1-x)^{m-1} dx$$

$$\leq \int_0^{1-\varepsilon} \frac{N_c!}{(N_c - m)!(m-1)!} x^{N_c-m}(1-x)^{m-1} dx = \mathcal{V}(N_c, m, \varepsilon)$$

where $\tau = \sup \{ x : F_v(x) < 1 - \varepsilon \}$. Therefore,

$$\Pr \left\{ \frac{\text{vol}_w \{ v(q) \leq \hat{v}_{N_c+1-m}, \ q \in \mathcal{Q}_C \}}{\text{vol}_w \{ \mathcal{Q}_C \}} \geq 1 - \varepsilon \right\} = \Pr \{ F_v(\hat{v}_{N_c+1-m}) \geq 1 - \varepsilon \}$$

$$= 1 - \Pr \{ F_v(\hat{v}_{N_c+1-m}) < 1 - \varepsilon \} \geq 1 - \mathcal{V}(N_c, m, \varepsilon).$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The equality holds if and only if \( \exists x^* \) such that \( F_u(x^*) = \varepsilon \) because such a \( x^* \) exists if and only if \( \tau = 1 - \varepsilon \). It follows that

\[
\Pr \left\{ \frac{\operatorname{vol}_w \{ u(q) \geq \hat{u}_m, q \in Q_C \}}{\operatorname{vol}_w \{ Q_C \}} \geq 1 - \varepsilon \right\} = \Pr \left\{ \frac{\operatorname{vol}_w \{ v(q) \leq \hat{v}_{N_c+1-m}, q \in Q_C \}}{\operatorname{vol}_w \{ Q_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \mathcal{V}(N_c, m, \varepsilon)
\]

with the equality holds if and only if \( \exists x^* \) such that \( F_u(x^*) = \varepsilon \).

The second part follows by applying Theorem 5.1 to the case of \( k = 1, i_1 = m \).

□

It is important to note that the two conditions in Theorem 5.2 are much weaker than the continuity assumption which requires that for any \( p \in (0,1) \) there exists \( x^* \) such that \( F_u(x^*) = p \). The difference is visualized in Figure 5.1. Cases A, B and E guarantee \( \exists x^* \) such that \( F_u(x^*) = \varepsilon \). Cases A, D and E guarantee \( \exists x^* \) such that \( \Pr \{ u(q) < x^* \mid q \in Q_C \} = 1 - \varepsilon \). Both conditions are violated in Case C. (The various magnitude of \( \varepsilon \) and \( 1 - \varepsilon \) is indicated by arrows at different heights.)

In general, it is important to know the probability of a quantity falling between two arbitrary samples. To that end, we have

**Corollary 5.2** Let \( 1 \leq m < n \leq N_c \). Suppose \( u(q) \neq \text{constant} \) in any open set of \( Q_C \). Then

\[
\Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_m < u(q) \leq \hat{u}_n, q \in Q_C \}}{\operatorname{vol}_w \{ Q_C \}} \geq 1 - \varepsilon \right\} = 1 - \mathcal{V}(N_c, N_c + 1 - n + m, \varepsilon).
\]

Since the condition that \( u(q) \neq \text{constant} \) in any open set of \( Q_C \) is equivalent to the absolute continuity assumption of \( F_u(x) \) (see the proof of Theorem 3.3 in [33]),
the proof of Corollary 5.2 can be completed by applying Theorem 5.1 to the case of $k = 2$, $i_1 = m$, $i_2 = n$ and $F_u(x)$ is continuous.

5.4 SAMPLE SIZE

The important issue of the randomized algorithms to robust analysis or optimal synthesis is to determine the minimum computational effort required to come up with a solution with a certain degree of 'reliability'. First, we consider this issue for the indirect approach.

5.4.1 CONSTRAINED SAMPLE SIZE

To estimate $\max_{Q_C} u(q)$ (or determine parameter $q$ achieving $\max_{Q_C} u(q)$), we have

**Corollary 5.3** Suppose that $\exists x^*$ such that $\Pr\{u(q) < x^* \mid q \in Q_C\} = 1 - \varepsilon$. Then

$$\Pr\left\{ \frac{\text{vol}_w \{u(q) \leq \hat{u}_{N_e}, q \in Q_C\}}{\text{vol}_w \{Q_C\}} \geq 1 - \varepsilon \right\} \geq 1 - \delta$$
if and only if \( N_c \geq \frac{\ln \frac{1}{\delta}}{\ln 1 - \varepsilon} \).

It should be noted that the results in Khargonekar and Tiku [33] and Tempo, Bai, and Dabbene [45] correspond to the sufficient part of the above Corollary for the special case of \( Q_C = Q \).

To estimate \( \min_{Q_C} u(q) \) (or determine parameter \( q \) achieving \( \min_{Q_C} u(q) \)), we have

**Corollary 5.4** Suppose that \( \exists x^* \) such that \( F_u(x^*) = \varepsilon \). Then

\[
\Pr\left\{ \frac{\text{vol}_w \{ u(q) \geq \hat{u}_1, q \in Q_C \}}{\text{vol}_w \{ Q_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \delta
\]

if and only if \( N_c \geq \frac{\ln \frac{1}{\delta}}{\ln 1 - \varepsilon} \).

To estimate the range of an uncertain quantity with a certain accuracy and confidence level apriori specified, we have the following corollary.

**Corollary 5.5** Suppose that \( u(q) \) \neq constant in any open set of \( Q_C \). Then

\[
\Pr\left\{ \frac{\text{vol}_w \{ \hat{u}_1 < u(q) \leq \hat{u}_{N_c}, q \in Q_C \}}{\text{vol}_w \{ Q_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \delta
\]

if and only if \( \mu(N_c) := (1 - \varepsilon)^{N_c - 1} [1 + (N_c - 1)\varepsilon] \leq \delta \).

Now we investigate the computational effort for the direct approach.

**5.4.2 Global Sample Size**

To estimate \( \min_{Q_C} u(q) \) (or determine parameter \( q \) achieving \( \min_{Q_C} u(q) \)), we have
Theorem 5.3 Suppose that \( \exists x^* \) such that \( F_u(x^*) = \varepsilon \). Then

\[
\Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_{\text{min}}, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \delta
\]

if and only if \( N \geq \frac{\ln(\frac{1}{\delta})}{\ln(1 - \varepsilon)} \).

Proof.

\[
\Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_{\text{min}}, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \right\} = \sum_{i=0}^{N} \Pr \{ M = i \} \Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_{\text{min}}, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \ | \ M = i \right\}
\]

\[
= \sum_{i=0}^{N} \binom{N}{i} \rho^i (1 - \rho)^{N-i} \Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_{\text{min}}, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \ | \ M = i \right\}.
\]

Notice that event

\[
\left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_{\text{min}}, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \ | \ M = i \right\}
\]

is equivalent to event

\[
\left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_1, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \ | \ L \leq N \right\}
\]

with \( N_c = i \) in the context of the indirect approach. By Corollary 5.1, we know that

\[
\Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_1, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \ | \ L \leq N \right\} = \Pr \left\{ \frac{\text{vol}_W \{ u(q) \geq \hat{u}_1, \ q \in Q_C \}}{\text{vol}_W \{ Q_C \}} \geq 1 - \varepsilon \right\}.
\]
Apply Theorem 5.2 to the case of $N_c = i$, $m = 1$, we have

$$\Pr \left\{ \frac{\text{vol}_w \{ u(q) \geq \hat{u}_1, q \in Q_C \} }{\text{vol}_w \{ Q_C \} } \geq 1 - \epsilon \right\} \geq 1 - \mathcal{V}(i, 1, \epsilon) = 1 - (1 - \epsilon)^i$$

with the equality holds if and only if $\exists x^*$ such that $F_u(x^*) = \epsilon$. Therefore

$$\Pr \left\{ \frac{\text{vol}_w \{ u(q) \geq \hat{u}_{\min}, q \in Q_C \} }{\text{vol}_w \{ Q_C \} } \geq 1 - \epsilon \right\} \geq \sum_{i=0}^{N} \left( \begin{array}{c} N \\ i \end{array} \right) \rho^i (1 - \rho)^{N-i} [1 - (1 - \epsilon)^i] = 1 - (1 - \epsilon \rho)^N$$

with the equality holds if and only if $\exists x^*$ such that $F_u(x^*) = \epsilon$. Finally, notice that $(1 - \epsilon \rho)^N \leq \delta$ if and only if $N \geq \frac{\ln(\frac{1}{\delta})}{\ln(\frac{1}{1-\rho})}$. This completes the proof. \qed

It should be noted that sufficiency part of the preceding theorem has been obtained in [33] in the context of estimating robust stability margin. By the similar argument as that of Theorem 5.3, we have the following result for estimating $\max_{Q_C} u(q)$ (or determine parameter $q$ achieving $\max_{Q_C} u(q)$).

**Theorem 5.4** Suppose that $\exists x^*$ such that $\Pr \{ u(q) < x^* \mid q \in Q_C \} = 1 - \epsilon$. Then

$$\Pr \left\{ \frac{\text{vol}_w \{ u(q) \leq \hat{u}_{\max}, q \in Q_C \} }{\text{vol}_w \{ Q_C \} } \geq 1 - \epsilon \right\} \geq 1 - \delta$$

if and only if $N \geq \frac{\ln(\frac{1}{\delta})}{\ln(\frac{1}{1-\rho})}$.

To estimate the range of a quantity for the system under a certain constraint $C$, we have
**Theorem 5.5** Suppose \( u(q) \neq \text{constant in any open set of } \mathcal{Q}_c \). Then

\[
\Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_{\text{min}} < u(q) \leq \hat{u}_{\text{max}}, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \right\} = 1 - \mu(N) \geq 1 - \delta
\]

if and only if \( \mu(N) := (1 - \varepsilon \rho)^{-1} [1 + (N - 1)\varepsilon \rho] \leq \delta \).

**Proof.**

\[
\begin{align*}
\Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_{\text{min}} < u(q) \leq \hat{u}_{\text{max}}, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \right\} &= \sum_{i=0}^{N} \Pr \{ M = i \} \Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_{\text{min}} < u(q) \leq \hat{u}_{\text{max}}, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \mid M = i \right\} \\
&= \sum_{i=0}^{N} \binom{N}{i} \rho^i (1 - \rho)^{N-i} \Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_{\text{min}} < u(q) \leq \hat{u}_{\text{max}}, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \mid M = i \right\}.
\end{align*}
\]

Notice that event \( \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_{\text{min}} < u(q) \leq \hat{u}_{\text{max}}, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \mid M = i \right\} \) is equivalent to event

\[
\left\{ \frac{\operatorname{vol}_w \{ \hat{u}_1 < u(q) \leq \hat{u}_i, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \mid L \leq N \right\}
\]

with \( N_c = i \) in the context of the indirect approach.

By Corollary 5.1 and Corollary 5.5, we have

\[
\begin{align*}
\Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_1 < u(q) \leq \hat{u}_i, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \mid L \leq N \right\} &= \Pr \left\{ \frac{\operatorname{vol}_w \{ \hat{u}_1 < u(q) \leq \hat{u}_i, \ q \in \mathcal{Q}_c \}}{\operatorname{vol}_w \{ \mathcal{Q}_c \}} \geq 1 - \varepsilon \right\} \\
&= 1 - (1 - \varepsilon)^{i-1} [1 + (i - 1)\varepsilon].
\end{align*}
\]
Therefore,

\[ \Pr \left\{ \frac{\text{vol}_w \{ \hat{u}_{\text{min}} \leq u(q) \leq \hat{u}_{\text{max}}, \; q \in \mathcal{Q}_C \}}{\text{vol}_w \{ \mathcal{Q}_C \}} \geq 1 - \varepsilon \right\} \]

\[ = \sum_{i=0}^{N} \binom{N}{i} \rho^i (1 - \rho)^{N-i} \left( 1 - (1 - \varepsilon)^{i-1} + (i-1) \varepsilon \right) \]

\[ = 1 - \sum_{i=0}^{N} \binom{N}{i} \rho^i (1 - \rho)^{N-i} (1 - \varepsilon)^{i-1} + (1 - \varepsilon)^i \]

\[ = 1 - \frac{1}{1 - \varepsilon} \sum_{i=0}^{N} \binom{N}{i} ((1 - \varepsilon) \rho)^i (1 - \rho)^{N-i} + \frac{\varepsilon}{1 - \varepsilon} \sum_{i=0}^{N} \binom{N}{i} ((1 - \varepsilon) \rho)^i (1 - \rho)^{N-i} \]

\[ -N \varepsilon \rho \sum_{i=1}^{N} \binom{N-1}{i-1} ((1 - \varepsilon) \rho)^{i-1} (1 - \rho)^{N-1-(i-1)} \]

\[ = 1 - \frac{1}{1 - \varepsilon} (1 - \varepsilon \rho)^N + \frac{\varepsilon}{1 - \varepsilon} (1 - \varepsilon \rho)^N - N \rho \varepsilon (1 - \varepsilon \rho)^{N-1} \]

\[ = 1 - (1 - \varepsilon \rho)^{N-1} [1 + (N-1) \varepsilon \rho] \]

\[ = 1 - \mu(N), \]

which implies that

\[ \Pr \left\{ \frac{\text{vol}_w \{ \hat{u}_{\text{min}} \leq u(q) \leq \hat{u}_{\text{max}}, \; q \in \mathcal{Q}_C \}}{\text{vol}_w \{ \mathcal{Q}_C \}} \geq 1 - \varepsilon \right\} \geq 1 - \delta \]

if and only if \( \mu(N) \leq \delta. \) \( \square \)
CHAPTER 6

CONCLUDING REMARKS

A fundamental concern of control engineer is the robust stability and performance for systems in the presence of uncertainties. This dissertation deals with the robustness analysis and synthesis problems in three different frameworks. First, the robustness of uncertain systems are discussed in the structured singular value $\mu$ framework. Parallel algorithms are developed which greatly facilitates robustness analysis. Second, the robust control problems are tackled in the Kharitonov framework. Efficient Algorithms have been developed for computing the robust $\mathcal{D}$-stability margin for arbitrary root domain $\mathcal{D}$. This allows for a more sophisticated analysis of system robustness. Finally, aimed at breaking through the barrier of NP hardness and reducing conservativeness, the robust control problems are considered in the probabilistic framework. Minimum computational effort for robust analysis with a certain degree of reliability is investigated and related sample sizes are derived. An interesting link between classic order statistics theory and robust control is established. Moreover, the classic order statistics distribution theory is generalized to accommodate discontinuous populations.

It is felt that further research can be done in the following directions.

- Computation of robust $\mathcal{D}$-stability margin with uncertainties bounded in a polytope: This is a much more general problem which includes the problems considered in Chapter 3 as special cases. A bounding techniques may be developed so that the Mapping Theorem can be applied. Also, the idea of Parallel Frequency Sweeping may be employed.
- Development of probabilistic branch and bound techniques: Chernoff bounds may be applied in the design of the algorithms. The idea of parallelism can also play an important role in improving the efficiency.

- Multiobjective analysis and synthesis: In general, such problems can not be solved in the deterministic framework. However, randomized algorithms may be the key to getting around the difficulties.
BIBLIOGRAPHY


[34] Lawrence, C. T., Tits, A. L. and Dooren, P. V., “A fast algorithm for the computation of an upper bound on the $\mu$-norm,” 1996 IFAC, pp. 59-64.


VITA

The author, Xinjia Chen, was born on September, 18, 1969, Canton, People's Republic of China. He received his bachelor degree of science in Electrical Engineering from Beijing University of Aeronautics and Astronautics, Beijing, China, in July 1991. From August 1991 to May 1992, he participated in the Social Education Program in Guangzhou. Since May 1992, he worked as a lecturer with Department of Wireless Communications, Guangzhou Teacher's University. He joined the Graduate Program at Louisiana State University in January, 1995, and is currently a candidate for the degree of Doctor of Philosophy, which will be conferred in December, 1999.
DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Xinjia Chen

Major Field: Electrical Engineering

Title of Dissertation: Probabilistic and Deterministic Algorithms for Information and Dynamic Systems

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

September 21, 1999