
UNIVERSITY OF
LIVERPOOL

# Analytic Function Methods for Nonparametric Control

Thesis submitted in accordance with the requirements of the

**University *of* Liverpool** for the

Degree of Doctor in Philosophy

by

Ming-Yen Chen

October 2013

# Statement of Originality

This thesis is submitted for the degree of Doctor in Philosophy in the Faculty of Engineering at the University of Liverpool. The research project reported herein was carried out, unless otherwise stated, by the author in the Department of Engineering at the University of Liverpool between 1/11/2009 and 31/10/2013.

No part of this thesis has been submitted in support of an application for a degree or qualification of this or any other University or educational establishment.

**Ming-Yen Chen**

**25th October 2013**

# Abstract

This thesis develops and investigates analytic function methods for nonparametric analysis and design of robust control linear systems. Compared to the parametric approaches, nonparametric approaches may enable the designer to directly use the experimental plant data to design the controller. Nonparametric approaches are potentially more accurate than parametric approaches since they do not need to make significant approximations due to parametric fittings. Moreover, since no parametric identification is required, nonparametric approaches are able to cope with time-delayed and differential difference systems. The design procedure process may also require less human judgement and so may be quicker and more readily automated. In this thesis, nonparametric approaches to control based on $\mathbb{H}^\infty$ analytic function theory is presented. It is the main purpose of this thesis to investigate the use of analytic function methods in $\mathbb{H}^\infty$ control problems. The implementation of the analytic methods and their applications are both addressed in the thesis.

In the $\mathbb{H}^\infty$ control approach, the controller achieving the stability requirement is synthesized to meet all the performance requirements in terms of an $\mathbb{H}^\infty$ norm. The $\mathbb{H}^\infty$ control problem is one where the controller and is designable in a system bounded by prescribed performances is generally viewed as a mathematical optimization problem. The methods to solve this optimization problem are required to find an optimizing solution functions that is analytic and bounded in the right-half complex plane. There are many existing control design approaches to the parametric representation of the problem, however, only a few methods exist for nonparametric $\mathbb{H}^\infty$ control. In this thesis, the nonparametric approaches based on the analytic function solutions to the $\mathbb{H}^\infty$ control problem are analyzed. The Disk Iteration method of Helton et al. [39], the Newton Iteration method of Helton and Merino[46], and the Linear Programming method of Streit [86] applied to control as suggested by Helton and Sideris [40] are implemented and examined :

- Disk Iteration (DI) Method : The theory of the DI method is summarized and an existing implementation due to Merino et al. [41] is translated

in Matlab language. Two nonparametric spectral factorization methods are also realized for the development of novel nonparametric control approaches using the DI method.

- Newton Iteration (NI) Method : The derivation of the NI method is outlined. A new implementation of the NI method in Matlab code is also presented for publication for the first time. In the NI method, the solution to an operator equation is obtained in terms of matrix representations of the operators. The difference between the performance of the DI method and the NI method is discussed and examined in several examples.

- Linear Programming (LP) Method : The LP method of Streit's algorithm [86] is implemented in Matlab for the first time. The interpolation method is replaced by Q-parameterization method to meet the internal stability requirement by a possibly nonparametric approach. An example is investigated and this illustrates the effectiveness of the LP method.

An comparison of the implementation of the three methods is made in the application to an engine control problem. The assessment of the resulting controllers is presented in terms of the time and frequency performance. The three methods are investigated in terms of the accuracy, computing power, and convergence.

In this thesis, it is concluded that, although the solution by the DI method is in general sub-optimal to the general optimization problem, this method is very effective and optimized for the circular form of optimization problems as well as quasi-circular type problems. The investigation of the NI method with several numerical examples indicates that not only are its solutions to the optimization problem optimal but also the NI method has a higher convergence rate for vector (i.e. multiple functions) cases. However, the NI method is very sensitive to the initial conditions. The study on the LP method shows that the LP method can be purely based on the measured frequency response data without the Fourier coefficient data but the method requires long computing time and large memory storage. An application to an automotive engine control problem by the proposed nonparametric analytic function methods is presented in this thesis and shows the effectiveness of this approach as an engineering control system design methodology.

# Acknowledgements

I'm particularly grateful to my supervisor Dr. Tom Shenton for his continuous support and patient guidance. His enthusiastic encouragement is highly appreciated in the completion of this work. I would also like to express my appreciation to Prof. Huajiang Ouyang as my second supervisor for his useful critiques and valuable advice during my PhD.

I would also like to thank many friends, Dr. Ahmed Abass, Dr. Shiyu Zhao, Dr. Ke Fang, Dr. Zhongyan Li, Kamil Ostrowski, Vincent Page and Dr. Hua Cheng for the fruitful discussions and helpful suggestions at different stages of the work throughout the years.

I would like to express my deep gratitude to my parents for their supportive love and inspiration. A special thank goes to my wife Shu-Hui Kuo for her unconditional support and patience.

# Contents

# List of Figures

# List of Tables

# Nomenclature

### Notations

| | |
|---|---|
| $[a, b]$ | Closed interval between $a$ and $b$ |
| $\bar{F}$ | Complex conjugate of the function $F$ |
| $\forall$ | For all |
| $\in$ | Belong to |
| $\mathbb{C}$ | The field of complex numbers |
| $\mathbb{C}^N$ | The ring of $N$ dimensional $\mathbb{C}$ space |
| $\mathbb{H}^1$ | The vector space (Hardy space) of complex-valued holomorphic functions $f$ bounded on the unit disk such that $\|f\|_{\mathbb{H}_1} = \sup_\theta \frac{1}{2\pi} \int_0^{2\pi} \left\|f\left(e^{j\theta}\right)\right\| d\theta < \infty$ |
| $\mathbb{H}^2$ | The vector space (Hardy space) of complex-valued holomorphic functions $f$ bounded on the unit disk such that $\|f\|_{\mathbb{H}_2} = \sup_\theta (\frac{1}{2\pi} \int_0^{2\pi} \left\|f\left(e^{j\theta}\right)\right\|^2 d\theta)^{1/2} < \infty$ |
| $\mathbb{H}_2(\partial D)$ | The set of the functions in $\mathbb{H}_2$ on the unit circle $\partial D$ |
| $\mathbb{H}_N^1$ | The ring of $N$-dimensional $\mathbb{H}^1$ space |
| $\mathbb{H}_N^{2\perp}$ | The orthogonal complement of the $N$-dimensional $\mathbb{H}^2$ space |
| $\mathbb{H}_N^2$ | The ring of $N$-dimensional $\mathbb{H}^2$ space |
| $\mathbb{RH}^\infty$ | The subspace of the real part of $\mathbb{H}^\infty$ where functions are real-valued, stable, and proper with real coefficients |
| $\mathbb{R}$ | The field of real numbers |
| $\mathbb{R}^+$ | The field of positive real numbers |

| | |
|---|---|
| $\mathbb{R}^N$ | The ring of $N$ dimensional $\mathbb{R}$ space |
| $\mathbb{H}^\infty$ | The vector space (Hardy space) of complex-valued holomorphic functions $f$ bounded on the unit disk such that $\|f\|_{\mathbb{H}_\infty} = \sup_\theta \left\|f\left(e^{j\theta}\right)\right\| < \infty$ |
| $\mathbb{H}_N^\infty$ | The ring of $N$-dimensional $\mathbb{H}^\infty$ space |
| $\mathcal{H}_a$ | Hankel operator with symbol $a$ |
| $\mathcal{L}_1$ | The Banach space of all measurable functions $f$ whose integral of the absolute value is finite, i.e. $\|f\|_{\mathcal{L}_1} \equiv \int_{-\infty}^{\infty} \|f\| \, d\omega < \infty$ |
| $\mathcal{L}_2$ | The Banach space of all measurable functions $f$ whose square root of integral of the square of the absolute values is finite, i.e. $\|f\|_{\mathcal{L}_2} \equiv (\int_{-\infty}^{\infty} \|f\|^2 \, d\omega)^{1/2} < \infty$. The $\mathcal{L}_2$ space is a *Hilbert* space. |
| $\mathcal{L}_2(\partial D)$ | The set of the functions in $\mathcal{L}_2$ on the unit circle $\partial D$ |
| $a.e.$ | Almost everywhere |
| $G$ | Transfer function of the plant |
| $K$ | Transfer function of the controller |
| $L$ | The open-loop transfer function defined as $L \equiv KG$ |
| $Q$ | The noise sensitivity function defined as $Q \equiv \frac{K}{1+KG}$ |
| $S$ | The primary sensitivity function defined as $S \equiv \frac{1}{1+KG}$ |
| $T$ | The complementary sensitivity function defined as $T \equiv \frac{KG}{1+KG}$ |
| $V$ | The disturbance sensitivity function defined as $V \equiv \frac{G}{1+KG}$ |
| $CT_a$ | Complex conjugate of Toeplitz operator with symbol $a$ |
| $Q^*$ | The optimal noise sensitivity function |
| $S^*$ | The optimal primary sensitivity function |
| $T'_{\alpha,\beta}$ | Jacobian operator with respect to the vector symbol $[\alpha, \beta]^T$ |
| $T^*$ | The optimal complementary sensitivity function |
| $T_a$ | Toeplitz operator with symbol $a$ |

| | |
|---|---|
| $V^*$ | The optimal disturbance sensitivity function |
| $A/B$ | The quotient space of a vector space $A$ by a subspace $B$ |
| $\partial D$ | The boundary of the unit circle |
| $\Re(F)$ | Real part of the function $F$ |
| $A^*$ | Complex conjugate of transpose of the matrix $A$ or the operator $A$ |
| $A^{-1}$ | Inverse of the matrix $A$ or the operator $A$ |
| $A^T$ | Transpose of the matrix $A$ |
| $D$ | The domain in the open unit circle |
| $I_n$ | $N \times N$ Identity matrix |
| $j$ | Square root of $-1$, i.e. $j = \sqrt{-1}$ |
| $s$ | Laplace transform variable |

**Abbreviations**

| | |
|---|---|
| DI | Disk Iteration method |
| ECU | Engine Control Unit |
| EMS | Engine Management System |
| L.H.P. | Left Half Plane |
| LFT | Linear Fractional Transform |
| LP | Linear Programming method |
| LTI | Linear Time Invariant |
| MBC | Model-Based Control |
| MIMO | Multiple-Input-Multiple-Output |
| NI | Newton Iteration method |
| NP | Nonparametric Method |
| PPP | Peak Pressure Position |
| QFT | Quantitative Feedback Theory |

| | |
|---|---|
| R.H.P. | Right Half Plane |
| SA | Spark Advance |
| SI | Spark-Ignition |
| SISO | Single-Input-Single-Output |
| SS | State-Space |
| T | Transpose |
| TDC | Top Dead Centre |
| TF | Transfer Function |
| TWC | Three Way Catalytic converter |
| UEGO | Universal Exhaust Gas Oxygen sensor |

# Chapter 1

# Introduction

Conventionally, to calibrate an engineering control system, the parametric identification of the system plant is needed in the first place in order to obtain its mathematical representation in the form of a model with a restricted (low) number of parameters. Based on the mathematical model, the controller is then designed or computed to meet the performance requirements specified by the designer. The controller is practically implemented in the real system generally as a feedback system and the performance of the controlled system is then examined and validated. This controller design approach is generally known as the Model-Based Control Method (MBC). However, the MBC method is, in practice, sensitive to the accuracy of the parametric model in the first system identification process. In addition, establishing a satisfactory model sometimes takes a long time and requires a priori knowledge. Controller design methods to potentially prevent a significant mismatch between the model and the real system can be achieved by what are known as the Nonparametric (NP) methods. They have been developed on the basis of the input-output plant data in either the time or frequency domain. Because NP methods are more directly based on the experimental data, the controller has the potential to be more robust to the system modelling errors. Moreover, most of the performance requirements, such as bandwidth, and gain-phase margin in the frequency domain are able to be considered in the frequency-based NP methods. NP approaches are therefore potentially powerful for designing robust controllers in engineering systems. Nevetheless, only a few NP control design methods are available to date, which accordingly motivates the research work in this thesis to investigate and develop such NP methods.

In the current industrial world, most complex systems are equiped with electrical control systems. In the automotive field, the Engine Control Unit (ECU) is a key component in an vehicle in terms of the engine performance. In a gasoline

engine, the ECU determines not only the amount of fuel and air intake into the engine chamber but also the spark timing to track the torque demand, the engine speed requirement, and control the other variables such as the temperature, emissions etc.. The control loops in the engine management system include those for most of the air-fuel ratio control, ignition control, idle speed control and knock control [35].

In this chapter, a literature review of feedback control theory is presented in Section 1.1. The $\mathbb{H}^\infty$ feedback theory is also outlined in details in this section. An introduction to the control loops in the automotive ECU are outlined in Section 1.2. Section 1.3 summarizes each chapter in the thesis. The contributions of the research work in this thesis are addressed in Section 1.4.

## 1.1 $\mathbb{H}^\infty$ Feedback Control

The controller design method formulated in terms of minimising the $\mathbb{H}^\infty$ norm is generally known as the $\mathbb{H}^\infty$ control problem and addresses the issues of the worst-case design for linear plants with unknown disturbances and unstructured plant uncertainties. The $\mathbb{H}^\infty$ space of functions represents the Hardy space of all functions which are analytic and bounded in the right half plane. The $\mathbb{H}^\infty$ controller design method was initially formulated by Zames [104] as a mathematical optimization problem in terms of the $\mathbb{H}^\infty$ norm. In the 1980s, theoretical work on approximation theory, functional analysis, operator theory and spectral factorization facilitated the optimal or sub-optimal controller designed by the $\mathbb{H}^\infty$ method in the frequency domain. The $\mathbb{H}^\infty$ control theory was reviewed and related to state-space means by Francis in [24] for finite-dimensional systems and by Foias in [23] for infinite-dimensional systems. The research in the time domain approach, which is based on the state-space representation of the system, motivated the development of $\mathbb{H}^\infty$ optimal control theory by means of Riccati equation solutions [21]. Many results and work by researchers have promoted the time domain $\mathbb{H}^\infty$ approach to deal with more general control problem, e.g. the time-variant control problem [6] and the nonlinear control problem[2].

Despite the fact that the state-space theory is relatively complete, the $\mathbb{H}^\infty$ control theory in frequency domain has several advantanges. One of the advantages of the frequency domain approach is that it can be deployed as a nonparametric $\mathbb{H}^\infty$ controller design method directly based on the output from an efficient Fast-Fourier-Transform-based computation of the frequency responses from the measured input-output data. Since the modelling inaccuracy may be less in the nonparametric method, this offers a possibly more accurate solution to the design

of a robust controller.

The frequency domain approach developed by Helton et al. [44, 47, 80] has been used in a sup-optimal nonparametric $\mathbb{H}^\infty$ controller design method for stable plants in [106]. Helton et al's theory is based on finding analytic function solutions to the Nevanlinna-Pick interpolation problem [109] and the related operator theory [5, 9, 77]. The sub-optimal nonparametric method was also successfully applied to the engine fuelling control system problem in [107, 108]. In terms of Helton et al's approach, the method aims to solve the mixed sensitivity control problem using appropriate weighting functions. However, the method is restricted to convex problems where the local optimum is the global optimum. Furthermore, the interpolation to guarantee the internal stability requires the parametric identification to allow computation of the Right-Half-Plane (RHP) poles and zeros.

Another nonparametric approach by means of parameterising the performance constraints in $\mathbb{H}^\infty$ norm was developed by Karimi and coworkers in [27] and summarized in [26, 56]. Karimi's method limits the selection of performance constraints in order to convexify the weighted sensitivity functions problem[56]. The application of his nonparametric method on a double-axis positioning system was presented in [57].

The other frequency domain approach of the Quantitative Feedback Theory (QFT) method was initiated by Horowitz [53]. This was adopted to a nonparametric method and applied to the automotive throttle control problem by A. Abass [1].

### 1.1.1 Feedback Control Theory

Two realizations of a *Linear-Time-Invariant* (LTI) physical system - the *transfer function* (TF) model and the *state-space* (SS) model - are introduced. Mathematically speaking, given the continuous input $u(t)$ and output $y(t)$, the transfer function $H(s)$ is defined as the relationship between the *Laplace* transform $\mathcal{L}\{\cdot\}$ of the input $u(s)$ and the output $y(s)$ with the complex variable $s$, as denoted

$$H(s) = \frac{\mathcal{L}\{y(t)\}}{\mathcal{L}\{u(t)\}} = \frac{Y(s)}{U(s)} \tag{1.1}$$

The Laplace transform maps a function in the time domain (e.g. $u(t)$ and $y(t)$) to the complex frequency domain . Mathematically, the substitution $s = j\omega$ connects the Laplace transform to Fourier transform. The transfer function model is possible to be estimated by using Fourier transform and is thereby commonly used in the *frequency-domain approach* of classical control theory.

Another representation of a $p \times q$ LTI system is the state-space model, which

describes the input-output relationship as a set of first order differential equations :

$$\begin{cases} \dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \\ y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \end{cases} \tag{1.2}$$

where $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^q$ are the $p$ inputs and $q$ outputs respectively, $x(t) \in \mathbb{R}^m$ are the $m$ state variables and $\mathbf{A}_{m\times m}$, $\mathbf{B}_{m\times p}$, $\mathbf{C}_{q\times m}$, $\mathbf{D}_{q\times p}$ are the system matrices.

The state-space model deals with functions in real time so the use of state space is viewed as the *time-domain approach* in the control theory. In addition, insight into the system (e.g. controllabilitiy, observability) is revealed by the state space realization even if the state variables are not related to physical meanings. The approach is also known to be numerically good for algorithms. As a result, there has been great interests in the development of the state-space approaches in recent years. Many applications in many control topics ( e.g. MIMO control problems, nonlinear control ) have been extensively used the state-space formulation [44, 83].

It is known that the purpose of control is to apply the control input to the plant to achieve desired output requirements. It is often straightforward to inverse the model that describes the dynamics of the plant system. In this case, the inverse transfer function can be used as the controller in a form of '*open-loop control*' as shown in Figure 1.1. The advantage of open-loop control is its simplicity and low cost to implement in real systems. Nevertheless, there exists several unavoidable issues with open-loop control, such as nonzero steady-state errors, possible instability when distrubances are present, non-robustness to plant parameter variations or errors, and the external disturbances [25]. As a result, the alternative strategy of closed-loop control or feedback control can be used to cope with the above problems.

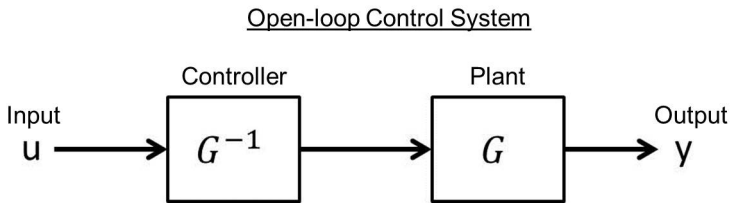



Figure 1.1: Structure of general open-loop control systems

The benefits of both the open and closed loop configurations may be obtained by the more general degree of freedom controller structure illustrated in Figure 1.2 where the output signal $y$ from the plant $G$ equipped with the con-

troller $K$ is fed to calculate the difference $e$ between the input signal $u'$ , which is pre-processed by the pre-filter $F$, and the output $y'$, which contains the disturbances $d$ and the noise $n$. The closed-loop control not only solves the problems of stability, robustness and reliability but also the capacity for more accurate tracking performance when disturbances are present. Nevertheless, compared to the open loop control scheme, the closed loop control system may be more complex and require more costs on additional sensors.

From Figure 1.2, the following variable relationships can be established in frequency domain

$$\begin{cases} u' = u \cdot F \\ e = u' - y \\ c = e \cdot K \\ c' = c + d \\ y' = c' \cdot G \\ y = y' + n \end{cases} \tag{1.3}$$

and formulated in the compact form :

$$y' = KGF\left[I + KG\right]^{-1} \cdot u + I\left[I + KG\right]^{-1} \cdot n + G\left[I + KG\right]^{-1} d \tag{1.4}$$

and

$$\begin{cases} e = F\left[I + KG\right]^{-1} \cdot u - G\left[I + KG\right]^{-1} \cdot d - I\left[I + KG\right]^{-1} \cdot n \\ c = KF\left[I + KG\right]^{-1} \cdot u - KG\left[I + KG\right]^{-1} \cdot d - K\left[I + KG\right]^{-1} \cdot n \end{cases} \tag{1.5}$$

The unitary closed-loop control system can be obtained with a prefilter $F = 1$. In this case, it can be observed from Equation 1.4 and 1.5 that the four sensitivity functions in the feedback control system are derived as

$$\begin{cases} S = I\left[I + KG\right]^{-1} = \left[I + L\right]^{-1} & \textit{Primary sensitivity function} \\ T = KG\left[I + KG\right]^{-1} = L\left[I + L\right]^{-1} & \textit{Complimentary sensitivity function} \\ Q = K\left[I + KG\right]^{-1} = K\left[I + L\right]^{-1} & \textit{Noise sensitivity function} \\ V = G\left[I + KG\right]^{-1} = G\left[I + L\right]^{-1} & \textit{Disturbance sensitivity function} \end{cases} \tag{1.6}$$

where $L = KG$ is called the *open-loop transfer function.*

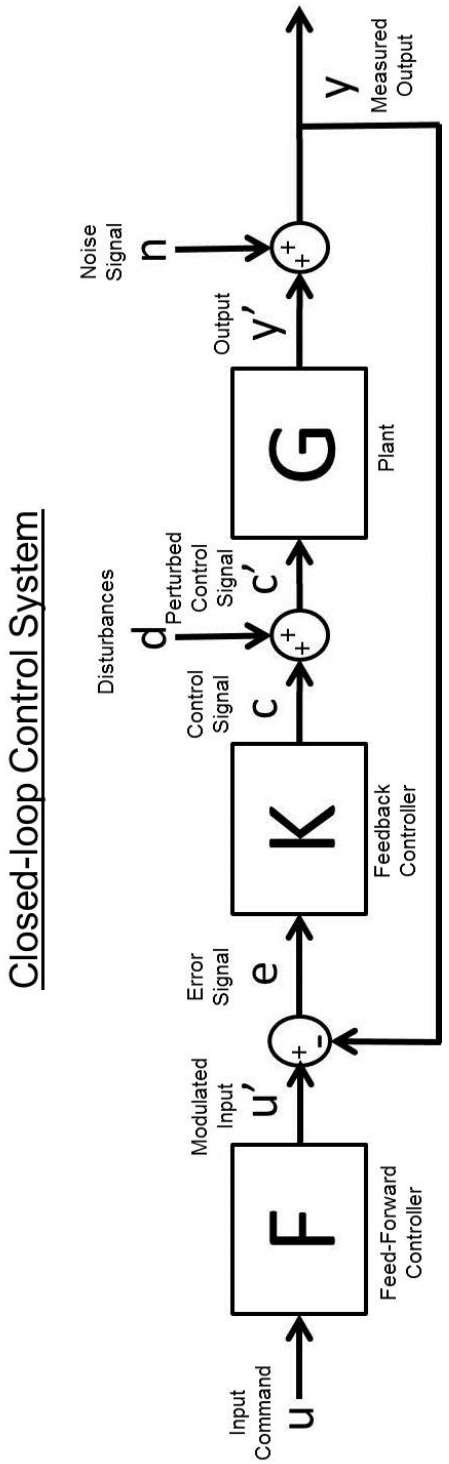



Figure 1.2: Standard structure of a two-degree-of-freedom feedback control system

These four sensitiviy functions are generally known as the *Gang of Four* [7], and they dominantly describe the dynamic characteristics of a control system. The primary sensitivity function $S$ defined as the transfer function from $u$ to $y'$. The complimentary sensitivity function $T$ defined as the function from the system's output $y$ to the input command $u$ provides the information about the tracking performance to the command $u$ in the system.

It is worth noting that the equation

$$S + T = I \tag{1.7}$$

is always true by the definitions of the functions $S$ and $T$ : $S = I\left[I + L\right]^{-1}$ and $T = L\left[I + L\right]^{-1}$ [1]. It is important to notice that Equation 1.7 limits the design of the controller naturally because, in Equation 1.7, a compromise between the functions $S$ and $T$ is obviously required. In other words, attempting to seek better tracking performance in $T$ results in the less robustness to the noise $n$ in $S$. The phenomenon is illustrated in terms of the corresponding Bode diagram of $S$ and $T$ in Figure 1.3.

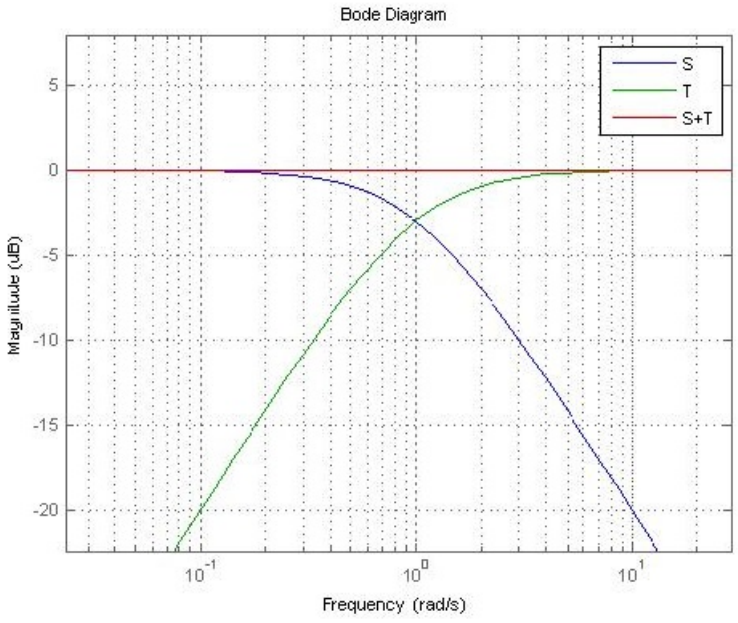



Figure 1.3: Bode diagram of the primary sensitivity function $S = \frac{2s+3}{2s^2+5s+3}$ and the complementary sensitivity function $T = \frac{2s^2+3s}{2s^2+5s+3}$

[1]

***REMARK*** In the multiple-input-multiple-output (MIMO) systems, the right hand side of Equation 1.7 is the *identity* matrix $I$. It is easy to observe that, in the single-input-single-output (SISO) systems, Equation 1.7 degenerates to the equation : $S+T = 1$.

### 1.1.2 Stability

The stability of the controlled system is a very important criterion in the controller synthesis since the unstable system may result in an unpredictable response. In the TF representation, it is known in the control theory [25, 62, 109] that if a linear continuous time-invariant system is to be *stable*, the roots of the closed-loop characteristic function $I + L(s) = 0$ ( i.e. the poles of the closed-loop system ) are all in the left half plane (L.H.P.). This is generally known as *Routh-Hurwitz stability criterion*. In the state space representation, the poles are usually computed as the roots of the characteristic equation $\det(sI - A) = 0$ where $s$ is the Laplace variable and $A$ is the matrix in Equation 1.2 [63].

In the feedback system shown in Figure 1.2, it is easy to derive from Equation 1.4 and 1.5 that, if no pole or zero in the right half plane ( R.H.P. ) of $G$ cancels out the zero or pole of $K$ in the R.H.P., the transfer functions $S$ and $T$ in Equation 1.6 must be stable [101]. That is, the system is internally stable *if and only if* the four sensitivity functions $S$, $T$, $Q$ and $V$ are all stable. This is summarized in the theorem [109]:

**Theorem 1.** *If there is no unstable pole-zero cancellation in $KG$ , the closed-loop system is internally stable if and only if one of the four sensitivity functions is stable.*

Simply speaking, the requirements for the internal stability conditions of a LTI feedback control system are expressed as [47]

- The complimentary sensitivity transfer function $T$ is in $\mathbb{RH}^\infty$
- No R.H.P. pole of the plant $G$ is cancelled by any R.H.P. zero of the controller $K$.
- No R.H.P. zero of the plant $G$ is cancelled by any R.H.P. pole of the controller $K$.

where $\mathbb{RH}^\infty$ here denotes the space of all rational transfer functions with real coefficients that have poles in L.H.P. ( i.e. stable )

### 1.1.3 Robustness

It is difficult to obtain an exact mathematical model of a physical plant due to the uncertainty in the system, the changes in plant dynamics, an error in system measurements and other unavoidable factors. Therefore, the study of modelling uncertainty is very important in control theories. Depending on different structure properties of the uncertainty, it is generally referred to either

*structured uncertainty* or *unstructured uncertainty*. The structured uncertainty is the representation of a known structure with some unknown perturbations in the parameters themselves, e.g. a percentage tolerance. The unstructured uncertainty, however, has less specific structure and is generally expressed as a global multiplier or addition of the gain at each frequnecy. The unstructured uncertainty is not accounted for in the system identification stage, especially in the high frequencies range. Details about representations of structured uncertainty and unstructured certainty are referred to several books and publications [20, 83, 109]. In this thesis, only unstructured uncertainty is considered.

Two common representations of the unstructured uncertainty are shown in Figure 1.4.The additive uncertainty structure shown in Figure 1.4a leads to the form of the perturbed plant $G = G_0 + \triangle$ , where $G_0$ is the nominal plant and $\Delta$ represents the difference between the real dynamics and the nominal model, i.e. the unmodelled synamics. In Figure 1.4b, the multiplicative uncertainty model is formulated in terms of $G$ as $G = G_0\,(I + \Delta)$. Both of the unstructured uncertainty models are commonly used and widely discussed in various papers and books [20, 83, 109]. The relationship between the unstructured uncertainty models and the conditions for the robustness is discussed in the following.

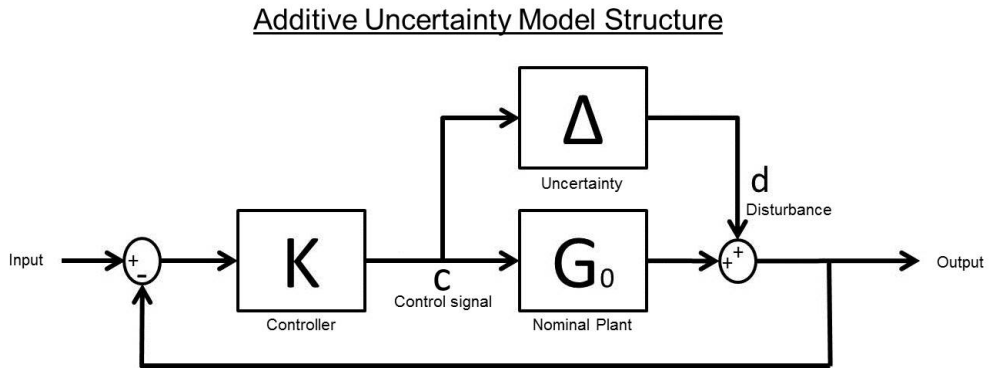
It is known that the stability condition of closed-loop systems is influenced by the uncertainty. The Small Gain Theorem by Zames [103], which is based on the feedback structure shown in Figure 1.5 where $G_1$ and $G_2$ are the LTI transfer functions, implies that the system performance is constrained to some extent by the uncertainty.

**Theorem 2.** *In Figure 1.5, if $G_1$ and $G_2$ are stable, the system is stable if*

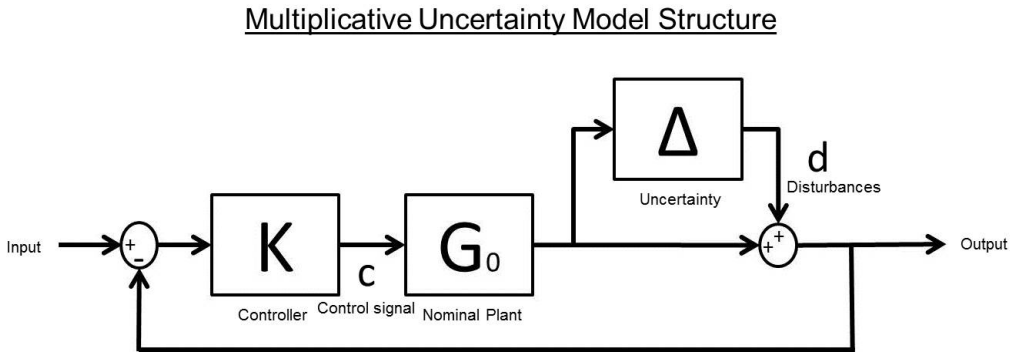
$$\|G_1 G_2\|_\infty < 1 \tag{1.8}$$

*and*

$$\|G_2 G_1\|_\infty < 1 \tag{1.9}$$




(a) Model structure of the additive uncertainty representation




(b) Model structure of the multiplicative uncertainty representation

Figure 1.4: Two typical representations of the uncertainty models

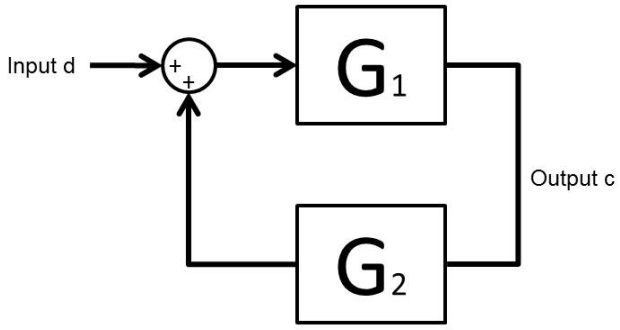



Figure 1.5: Positive Feedback Control to explain Small Gain Theorem

Consider the structure of the feedback control system with the additive uncertainty model in Figure 1.4a, we can easily derive the transfer function from signal $c$ to signal $d$ as

$$T_{cd} = K\left[I + KG\right]^{-1}$$

In other words, the uncertaint system in Figure 1.4a is equivalent to the sytem in Figure 1.6.

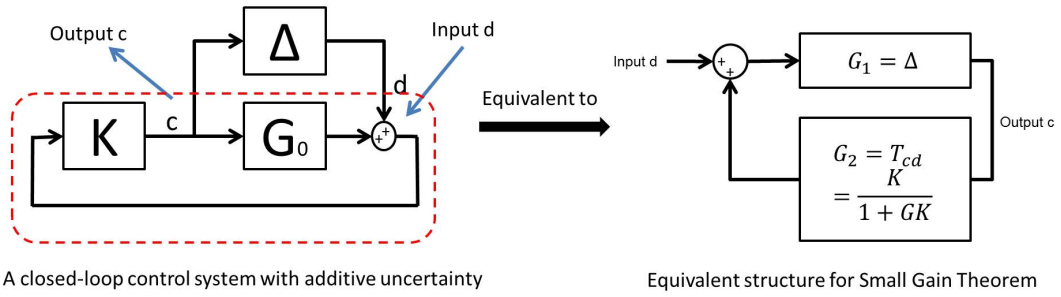



Figure 1.6: Equivalent feedback control system for the Small Gain Theorem

Let $G_1 = \triangle$, $T_{cd}$ is viewed as $G_2$ in Figure 1.5. By the *Small Gain Theorem*, we then have

**Theorem 3.** *For a stable uncertainty set $\triangle$, the closed loop system is stable if*

$$\left\|\triangle \cdot K\left[I + KG\right]^{-1}\right\|_\infty < 1 \text{ and } \left\|K\left[I + KG\right]^{-1} \cdot \triangle\right\|_\infty < 1$$

i.e.

$$\left\|K\left[I + KG\right]^{-1}\right\|_\infty < \frac{1}{\|\triangle\|_\infty} \tag{1.10}$$

Writing the formulation of the additive uncertainty as $\triangle = \bar{\triangle} \cdot W$ where $\bar{\triangle}$ is the normalized norm of the perturbation ( i.e. $\left\|\bar{\triangle}\right\|_\infty \leq 1$ ) and $W$ is the

weighting function on $\bar{\triangle}$, we can reform Inequality 1.10 as

$$\left\|W \cdot K\left[I + KG\right]^{-1}\right\|_\infty = \left\|W \cdot Q\right\|_\infty < \frac{1}{\left\|\bar{\triangle}\right\|_\infty} = 1 \tag{1.11}$$

For the multiplicative uncertainty model, by the *Small Gain Theorem*, the stability condition for the feedback control system becomes

$$\left\|KG\left[I + KG\right]^{-1}\right\|_\infty < \frac{1}{\left\|\triangle\right\|_\infty} \tag{1.12}$$

i.e.

$$\left\|W \cdot KG\left[I + KG\right]^{-1}\right\|_\infty = \left\|W \cdot T\right\|_\infty < \frac{1}{\left\|\bar{\triangle}\right\|_\infty} = 1 \tag{1.13}$$

The inequalities 1.11 and 1.13 are known as *Robust Stability* (RS) conditions. Table 1.1.1 lists the *Robust Stability* (RS) conditions for different types of uncertainty models. The derivations of these types of uncertainty models are well known in many texts e.g. [18]

| Uncertainty Plant $\mathbf{G}_p$ | Robust Stability (**RS**) conditions |
|---|---|
| $\mathbf{G_p} = G \cdot [1 + \triangle \cdot W]$ | $\left\|W \cdot \frac{KG}{I+KG}\right\|_\infty = \left\|W \cdot T\right\|_\infty \leq 1$ |
| $\mathbf{G_p} = G + \triangle \cdot W$ | $\left\|W \cdot \frac{K}{I+KG}\right\|_\infty = \left\|W \cdot Q\right\|_\infty \leq 1$ |
| $\mathbf{G_p} = G \cdot [I + \triangle \cdot W \cdot G]^{-1}$ | $\left\|W \cdot \frac{G}{I+KG}\right\|_\infty = \left\|W \cdot V\right\|_\infty \leq 1$ |
| $\mathbf{G_p} = G \cdot [1 + \triangle \cdot W]^{-1}$ | $\left\|W \cdot \frac{I}{I+KG}\right\|_\infty = \left\|W \cdot S\right\|_\infty \leq 1$ |

Table 1.1.1: The RS conditions for different types of uncertainty

### 1.1.4 $\mathbb{H}^\infty$ Control

The robustness is an important topic in the control system design. A well designed control system often results in good command tracking performance despite the unmodelled dynamics and the environmental disturbances. Robustness has been a dominantly influential topic in system control theory in the 1960s and 1970s. $\mathbb{H}^\infty$ optimal control theory was then developed in the early 1980s when Zames [104] considered a control problem as an optimization in terms of $\mathbb{H}^\infty$ norm to minimize the disturbance insensitivity in feedback systems. Since then, many efforts on developing the $\mathbb{H}^\infty$ control theory were made by researchers. The theory was soon recognized to deal with more constraints, such as robustness to modelling errors and other performance requirements. The $\mathbb{H}^\infty$ method developed by Helton et al. is systematic and extensible to more general control problems with classical control theory [47]. The extension of the $\mathbb{H}^\infty$ control theory to nonlinear systems has also been an active research topic in optimal control theory

in recent years. Several textbooks [24, 83, 109] introduce the theory of the $\mathbb{H}^\infty$ method as well as their applications. In this section, the background of the $\mathbb{H}^\infty$ optimal control theory is described.

In general, the guidelines to design a robust control system are described as follows :

For the closed-loop system in Figure 1.7, in terms of the sensitivity functions $S, T, U$, it is claimed that

- For the purpose of *disturbance rejection*, $S$ to be small.
- For the purpose of *noise attenuation*, $T$ to be small.
- For the purpose of *reference tracking*, $T$ to be close to identity.
- For the purpose of *control effort minimization*, $Q$ to be small.
- For the purpose of maintaining *robust stability*, $Q$ to be small for the additive uncertainty model or $T$ to be small for the multiplicative uncertainty model.

It is observed that the requirements for disturbance rejection and noise attenuation are in conflict because Equation 1.7 : $S + T = 1$ must be satisfied. This requires a trade-off which is fundamental to closed loop systems and the trade-off is generally achieved by making $T$ small at high frequency and $S$ small at low frequency, as shown in Figure 1.3. Moreover, to accomplish the reduction of the disturbances in terms of the sensitivity function $S$, the weighting function $W_S^{-1}$ is selected to reflects the desired shape of the function $S$. It is in general required the function $S$ is low at low frequencies as shown in Figure 1.8a. On the other hand, since the complimentary sensitivity function $T$ determines the tracking performance and the robustness characteristic, the weighting function $W_T^{-1}$ is chosen to shape the function $T$ to keep it near unity at low frequencies and as low as possible to reject the noise at high frequencies in Figure 1.8b. It is therefore known that these requirements on $S$ and $T$ (and sometimes on $Q$) can be viewed as the design specifications equivalent to the constraints of the design boundary.

It is further discussed that, assuming that the multiplicative uncertainty model represents the perturbation in the plant model $G$, the robust stability condition leads to the first constraint

$$\left\| W_T \cdot KG \left[ I + KG \right]^{-1} \right\|_\infty = \left\| W_T \cdot T \right\|_\infty < 1 \tag{1.14}$$

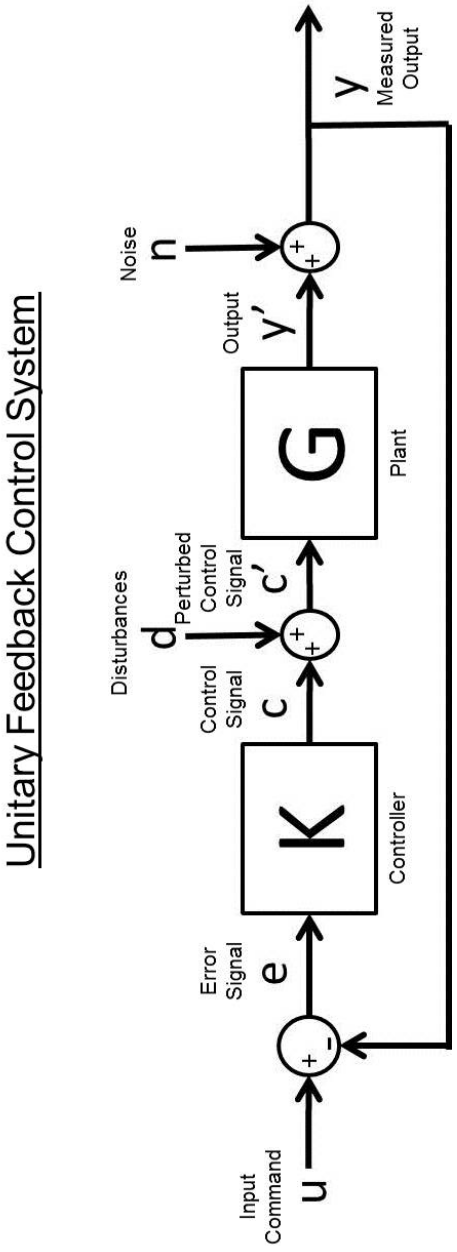



Figure 1.7: Structure of a unitary feedback control system

i.e.

$$\|T\|_\infty < |W_T|^{-1}, \forall\omega \tag{1.15}$$

Thus, it can be known that the peak value of the function $T$ is bounded by $|W_T|^{-1}$.

In addition, another important property of a closed loop system is the nominal performance (**NP**) requirement of the sensitivity function $S$ [83]

$$\|W_S \cdot S\|_\infty < 1, \forall\omega \tag{1.16}$$

i.e.

$$\|S\|_\infty < |W_S|^{-1} \tag{1.17}$$

This implies that the magnitude of $S$ is bounded by $|W_S|^{-1}$. Typical representations of the two constraints (**RS** and **NP**) are plotted in Figure 1.8a and Figure 1.8b respectively where the two weighting functions $W_S$ and $W_T$ are viewed as the upper bounds for $S$ and $T$. The controlled system performance can thus be tailored by properly choosing the weighting functions $W_T$ and $W_S$ so as to achieve the desired performance.

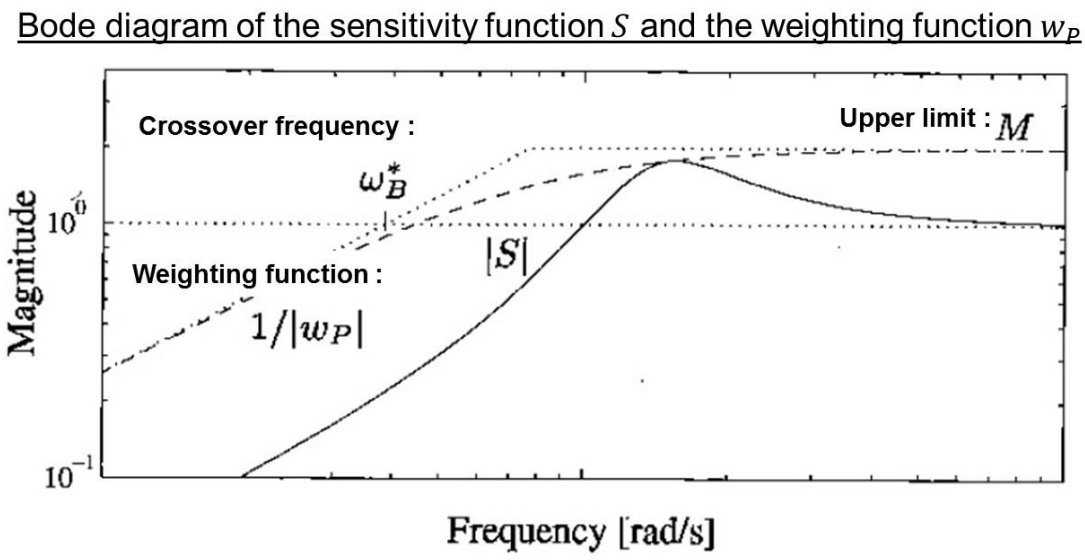
The weight shaping problem is generally acknowledged as the key requirement to success in the $\mathbb{H}^\infty$ control problem. The requirements in Equation 1.14 and 1.16 may be expressed in the stacked form as the so-called mixed sensitivity problem

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty < 1 \tag{1.18}$$

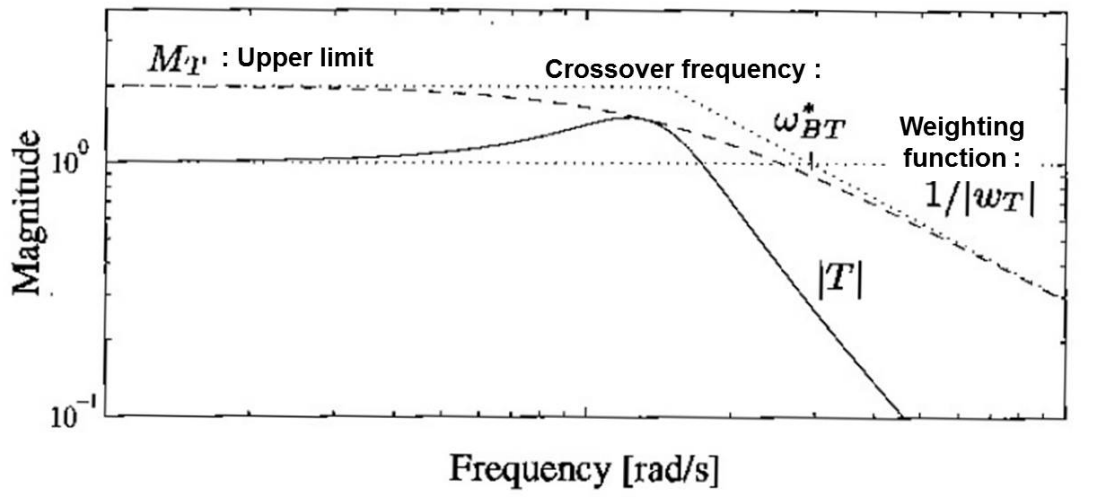
The minimization of the $\mathbb{H}^\infty$ norm of the mixed sensitivity problem is addressed :
By rewriting $W_S$ and $W_T$ as a function of $\gamma$,

$$\gamma = \min_{Stabilizing \quad K} \left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \tag{1.19}$$

Moreover, not only the bounds $W_S$ and $W_T$ for $S$ and $T$ but also the bound $W_Q$ for $Q$ may be included. The problem 1.19 can thus be extended to




(a) $S$ and $1/W_s$




(b) $T$ and $W_T$

Figure 1.8: Bode diagrams of the relationship between the two sensitivity functions $S$ and $T$ and the weighting function $W_S$ and $W_T$[83]

$$\gamma = \min_{Stabilizing \quad K} \left\| \begin{bmatrix} W_S S \\ W_T T \\ W_Q Q \end{bmatrix} \right\|_\infty \tag{1.20}$$

This is generally known as the canonical form of the mixed sensitivity control problem[2]. The derivation of Equation 1.19 is presented below.

Consider the system in Figure 1.9 and compare it with Figure 1.7, $G$ is the augmented plant model that consists of the nominal plant model and the uncertainty model $\Delta$, $K$ is the controller, the exogenous output from the system is $z$, and the exogenous input signals $w$ includes the input command $u$, disturbance signal $d$ and the noise $n$ .

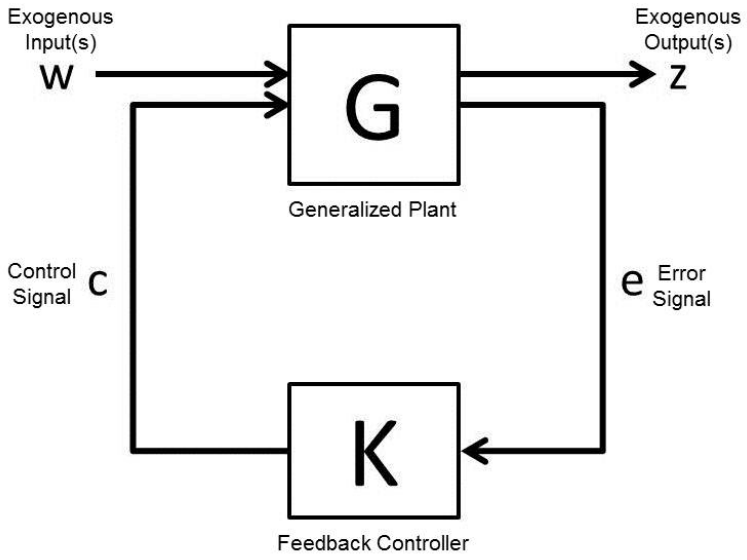



Figure 1.9: General model structure to form $\mathbb{H}^\infty$ control problem

[2]

***REMARK*** *For mathematical convenience, the stacked specifications provides an intuitive way to combine all the requirements. However, the procedure in fact sacrifices the accuracy of each specification. For example, if the Euclidean norm is used in the mixed sensitivity problem, we can rewrite Equation 1.18 as*

$$\left\| \begin{bmatrix} W_S \cdot S \\ W_T \cdot T \end{bmatrix} \right\|_\infty = \max \sqrt{|W_s \cdot S|^2 + |W_T \cdot T|^2} < 1$$

*This is very similar to Equation 1.14 and 1.16. However, suppose $|W_S \cdot S| = |W_T \cdot T|$ in the worst case, we have $|W_S \cdot S| \leq 0.707$ and $|W_T \cdot T| \leq 0.707$. Compared to the Equation 1.14 and 1.16, these two constraints are more stringent by a factor of $1/\sqrt{2}$. However, because the choice of the weighting functions is flexible, this can be taken into account in selecting the weighting functions. Therefore, the use of the Euclidean norm still useful in the $\mathbb{H}^\infty$ control.*

In Figure 1.9, the control laws for such system are expressed as

$$\begin{bmatrix} z \\ e \end{bmatrix} = G \begin{bmatrix} w \\ c \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix}$$

and

$$c = Ke$$

The transmission function $F_{zw}$ between the exogenous output signal $z$ and the the exogenous input signal $w$ is derived as

$$z = \left(G_{11} + G_{12}K\left(I - G_{22} \cdot K\right)^{-1} G_{21}\right) w = F_{zw}w \tag{1.21}$$

In terms of the system in Figure 1.10, Equation 1.21 becomes

$$\begin{bmatrix} z_S \\ z_T \\ z_U \\ e \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u \\ c \end{bmatrix} \tag{1.22}$$

where

$$G_{11} = \begin{bmatrix} W_S \\ 0 \\ 0 \end{bmatrix}, \quad G_{12} = \begin{bmatrix} -W_S G \\ W_Q \\ W_T G \end{bmatrix}, \quad G_{21} = I, \quad G_{22} = -G$$

The transmission function $F_{zw}$ is then derived as

$$F_{zw} = G_{11} + G_{12}K\left(I - G_{22}K\right)^{-1} G_{21} = \begin{bmatrix} W_S\left[I + GK\right]^{-1} \\ W_Q K\left[I + GK\right]^{-1} \\ W_T GK\left[I + GK\right]^{-1} \end{bmatrix} = \begin{bmatrix} W_S S \\ W_Q Q \\ W_T T \end{bmatrix} \tag{1.23}$$

The term in Equation 1.23 may then be used to formulate the standard $\mathbb{H}^\infty$ mixed sensitivity problem as

$$\gamma = \min_{Stabilizing \quad K} \|F_{zw}\|_\infty = \min_{Stabilizing \quad K} \left\| \begin{bmatrix} W_S S \\ W_T T \\ W_Q Q \end{bmatrix} \right\|_\infty$$

where $\gamma$ is a positive real number.

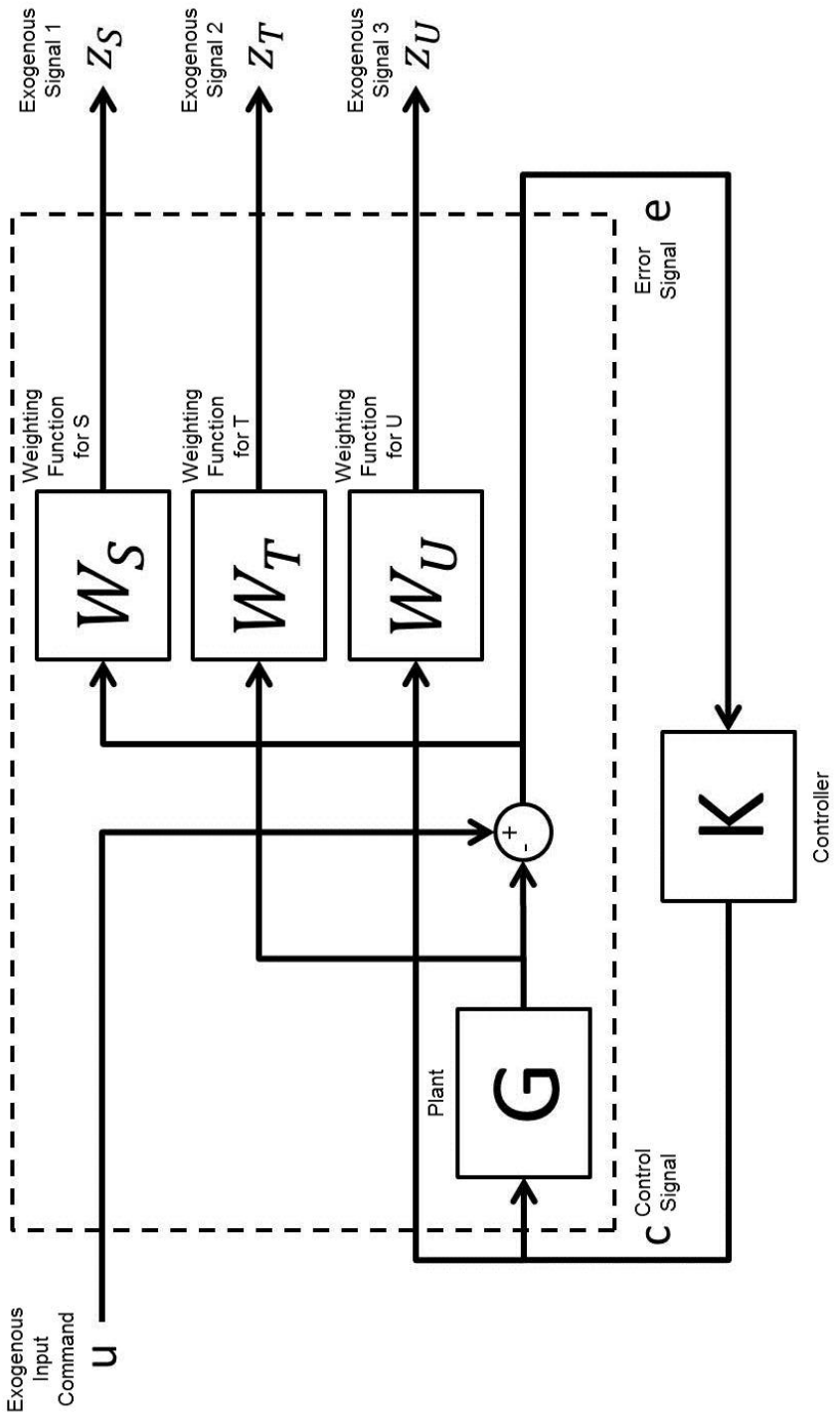



Figure 1.10: Standard structure of the $\mathbb{H}^\infty$ mixed sensitivity problem

The algorithm proposed by Doyle et al. [21] to solve the above problem is summarized :

**Theorem 4.** *In the standard form of a feedback control system in Figure 1.9, for the plant $G$ with the state space realization of*

$$\begin{cases} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 c(t) \\ z(t) &= C_1 x(t) + D_{11} w(t) + D_{12} c(t) \\ e(t) &= C_2 x(t) + D_{21} w(t) + D_{22} c(t) \end{cases}$$

*i.e. the generalized plant is* $G = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}$

*with the following assumptions :*

- $(A, B_2, C_2)$ *is stabilizable and detectable*
- $D_{12}$ *and* $D_{21}$ *have full rank*
- $\begin{bmatrix} A - j\omega I & B_2 \\ C_1 & D_{12} \end{bmatrix}$ *has full column rank for all* $\omega$
- $\begin{bmatrix} A - j\omega I & B_1 \\ C_2 & D_{21} \end{bmatrix}$ *has full row rank for all* $\omega$
- $D_{11} = 0$ *and* $D_{22} = 0$

*there exists a stabilizing controller $K$ such that, for a given positive number $\gamma$,*

$$\|F_{zw}\|_\infty < \gamma$$

*if and only if*

1. *the solution $X_\infty \geq 0$ to the algebraic Riccati equation*

$$A^T X_\infty + X_\infty A + C_1^T C_1 + X_\infty \left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\right) X_\infty = 0$$

*is such that*

$$Re\left(\lambda_i \left[A + \left(\gamma^{-2} B_1 B_1^T - B_2 B_2^T\right) X_\infty\right]\right) < 0, \forall i$$

2. *the solution* $Y_\infty \geq 0$ *to the algebraic Riccati equation*

$$AY_\infty + Y_\infty A^T + B_1 B_1^T + Y_\infty \left(\gamma^{-2} C_1^T C_1 - C_2^T C_2\right) Y_\infty = 0$$

*is such that*

$$Re\left(\lambda_i \left[A + Y_\infty \left(\gamma^{-2} C_1 C_1^T - C_2 C_2^T\right)\right]\right) < 0, \forall i$$

*the spectral radius* $\rho\left(X_\infty Y_\infty\right) < \gamma^2$

The solution to the $\mathbb{H}^\infty$ feedback controller is thus given by the solutions to these two algebraic *Riccati equations* above by [83]

$$K(s) = -F_\infty \left(sI - A_\infty\right)^{-1} Z_\infty L_\infty$$

where

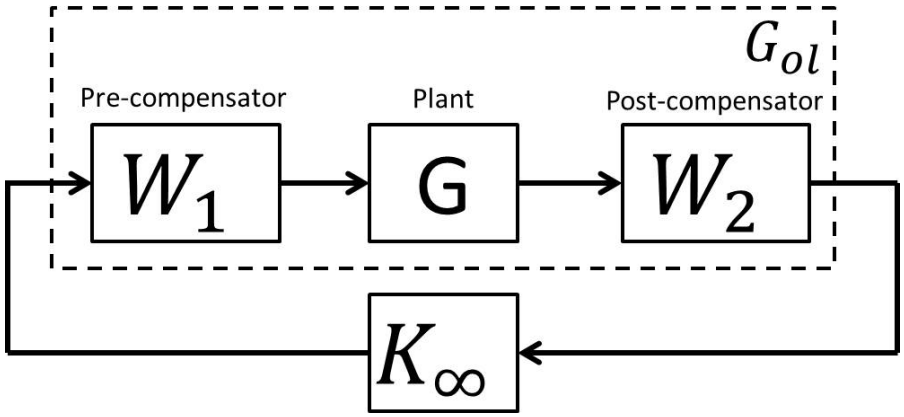
$$\begin{aligned} F_\infty &= -B_2^T X_\infty \\ Z_\infty &= \left(I - \gamma^{-2} Y_\infty X_\infty\right)^{-1} \\ L_\infty &= -Y_\infty C_2^T \\ A_\infty &= A + \gamma^{-2} B_1 B_1^T X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2 \end{aligned}$$

The program codes to implement the solution to the two *Riccati equations* are implemented in *MATLAB Robust Control Toolbox* [8], which reliably and efficiently compute the solution to the Riccati equations for the $\mathbb{H}^\infty$ controller.

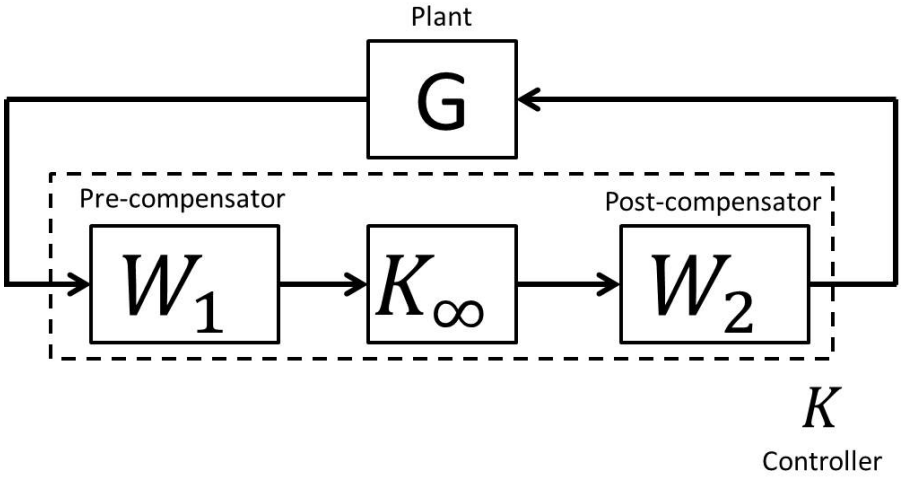
In addition to the above $\mathbb{H}^\infty$ approach based on solving the Riccati equations, an alternative approach proposed by McFarlance and Glover [30, 66, 67] is known as $\mathbb{H}^\infty$ *loop-shaping method*, which is based on the $\mathbb{H}^\infty$ robust stabilization by the classical loop shaping technique and has the great advantage that weighting functions do not need to be chosen. The design process of the $\mathbb{H}^\infty$ loop-shaping method is separated into two stages. Firstly, the open-loop plant $G_{ol}$ equipped with a pre-compensator $W_1$ and a post-compensator $W_2$ as shown in Figure 1.11a is given by [83]

$$G_{ol} = W_1 G W_2$$

The robust stabilization problem which is then addressed is to find the stabilizing $\mathbb{H}^\infty$ controller such that




(a) The $\mathbb{H}^\infty$ loop shaping procedure-1




(b) The $\mathbb{H}^\infty$ loop shaping procedure-2

Figure 1.11: Control structures in the $\mathbb{H}^\infty$ loop shaping method

$$\left\| \begin{bmatrix} I \\ K_\infty \end{bmatrix} (I - G_{ol}K_\infty)^{-1} M^{-1} \right\|_\infty \leq \epsilon^{-1}$$

where $\epsilon$ is selected such that $\epsilon \leq \epsilon_{max}$ where $\epsilon_{max}$ is defined as

$$\epsilon_{max}^{-1} \triangleq \inf \left\| \begin{bmatrix} I \\ K_\infty \end{bmatrix} (I - G_{ol}K_\infty)^{-1} M^{-1} \right\|_\infty$$

and $G_{ol} = M^{-1}N$ where $M, N \in \mathbb{H}^\infty$ are the normalized co-prime factors of $G_{ol}$ and satisfies the Bezout identity equation

$$M \cdot M^* + N \cdot N^* = I$$

Secondly, the implemented feedback controller $K$ is then computed by combining the two weighting functions $W_1$ and $W_2$ with $K_\infty$ as shown in Figure 1.11b by

$$K = W_1 K_\infty W_2$$

In contrast to the above state-space approaches, Helton et al.[47] proposed a systematic method to solve the $\mathbb{H}^\infty$ control problem in the frequency domain. The $\mathbb{H}^\infty$ optimization problem in his approach mathematically views
Equation 1.20 as a *minimax* optimization problem. Such optimization problem can be written as

$$\begin{aligned} \min \Gamma(\omega, f(j\omega)) &= \min_{Stabilizing K} \left\| \begin{bmatrix} W_S S \\ W_T T \\ W_U \cdot U \end{bmatrix} \right\|_\infty && (1.24) \\ &= \min_{Stabilizing K} \max_{j\omega} (W_S S, W_T T, W_U U) && (1.25) \\ &\leq \gamma^* && (1.26) \end{aligned}$$

where $\Gamma(\omega, f(j\omega))$ is the objective function which is dependant on the variables $f(j\omega)$ and $\omega$ , and $f(j\omega)$ is any analytic function in the frequency domain (e.g. $S$, $T$ or $Q$).

Helton and Merino [47] proposed that the optimal solution to Equation 1.24 (the $f$ term in Equation 1.24 is usually the primary sensitivity function $T$ or the noise sensitivity function $Q$) provides the solution of the stabilizing controller (which can be computed by $K = T[(I - T)G]^{-1} = Q[I - GQ]^{-1}$). The controller may be identified as a high-order rational transfer function. However, normally, the rational function of the controller solution is required to be of low order to

be implemented in the real system. This can be accomplished in the system identification process and by using model reduction techniques [29, 81].

Furthermore, the objective function $\Gamma(\omega, f(j\omega))$ in Equation 1.24 is often converted to the function $\Gamma\left(e^{j\theta}, f\left(e^{j\theta}\right)\right)$ on the unit circle. This is because the optimization problem in the $e^{j\theta}$ domain can be treated as a Nevanlinna-Pick interpolation problem [71, 73] and solved by using the solution to the Nehari problem [70]. For scalar cases, the solution to the Nehari problem is known to be unique and available by [3, 4, 5]. For vector cases, the Nehari problem can be solved by Nehari commutant lifting formula [47, 69, 77]. As a result, Helton and Merino's method attempts to approximate the performance function $\Gamma$ in Equation 1.24 to the quadratic form using the solution update $f^{k+1} = f^k + th$ given by

$$\Gamma\left(z, f^{k+1}\right) = g^k + \sum_{i=1}^{N} 2 \cdot Re\left(a_i^k h_i^k\right) + \sum_{i=1}^{N} \delta \left|h_i^k\right|^2 \tag{1.27}$$

where $z$ is the set of sampling points on the unit circle, and $g^k = \Gamma\left(z, f^k\right)$, $a^k = \frac{\partial}{\partial z}\Gamma\left(z, f^k\right)$, $\delta = \frac{\partial^2}{\partial z \partial \bar{z}}\Gamma\left(z, f^k\right)$
The function $\Gamma$ in Equation 1.27 by iterating to the closest circular form is then related to the Nehari problem. The solution to the $\mathbb{H}^\infty$ optimization problem is then possibly obtained. Then this sub-optimal solution is transformed to the frequency domain to sythesize the rational function of the controller. The algorithm and the approach for solving the $\mathbb{H}^\infty$ problems is coded as a package in Mathematica language and is available for academic use at [41].

Another algorithm proposed by Helton et al. [39] is based on the Newton Iteration (NI) method to solve the operator equations for optimality conditions. The solution is optimal to the original optimization problem and converges very quickly. However, the method is found to be sensitive to the initial points and may diverge as the iteration continues.

The two algorithms to solve the problem are therefore discussed respectively in more details in Chapter 2 and 3.

To take advantage of the fact that frequency responses may be used almost directly from the experimental data, Zhao et al [107] developed a sub-optimal nonparametric $\mathbb{H}^\infty$ approach based on Helton's approach [47] and applied this to an automotive engine Peak-Pressure-Position ( P.P.P. ) control problem. In the extension of Zhao et al's frequency approach to MIMO systems, a norm called $\mathbb{H}^\infty$ Frobenius norm in [92], defined as
$\sup_{\omega} \|G(j\omega)\| = \sup_{\omega} \left(trace(G(-j\omega) \cdot G(j\omega))\right)^{1/2}$ where $G(j\omega)$ is the frequency response function, is introduced. Although the NI method cannot be used to

optimise MIMO systems using the maximum singular value norm, it can be used for the $\mathbb{H}^\infty$ Frobenius norm for MIMO systems which has itself certain advantages[92]. The $\mathbb{H}^\infty$ Frobenius norm optimization approach can be adapted to solve a form of the mixed sensitivity control problem, which allows better control over the elements of the closed loop transfer function matrices, such as allowing decoupling[108]. In particular, Helton's method was extended by Zhao [106] as sub-optimal nonparametric method to deal with M.I.M.O. control problems and also implementeed on the application of the Torque-$\lambda$ decoupling control in the Engine Control Unit (ECU) [108].

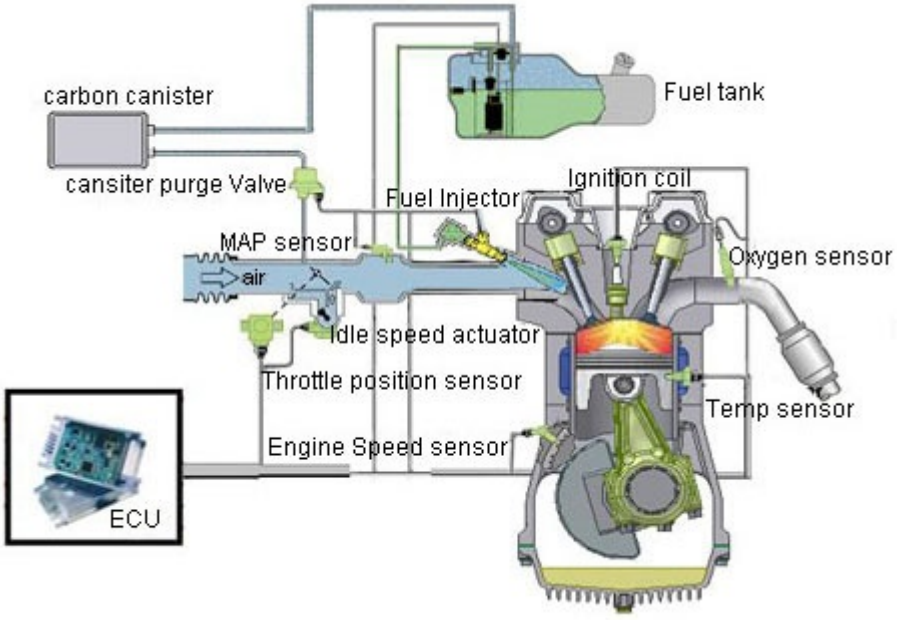
## 1.2 Automotive Engine Control

The engine is the key part to generate the power to accelerate automotive vehicles. Nowadays, most vehicles are equipped with internal combustion (IC) engine because its superior thermal efficiency and compactness of the systems. In the recent engine technology, gasoline and diesel engines are the most commonly used types of internal combustion engine systems in terms of different thermal cycles and fuel properties. The gasoline engine is degisned by means of the Otto cycle, which based on the four strokes : intake, compression, combustion and exhaust. The fuel in the gasoline engine is pre-mixed in the intake manifold and ignited by the spark plug in the combustion chamber. As a result, the gasoline engine is also known as the spark ignition (SI) engine. The diesel engine, however, is built according to Diesel cycle. The combustion in diesel engine is based on the self-ignition of the fuel. There is no spark plug in diesel engine but the engine itself has the same four strokes as the gasoline engine.

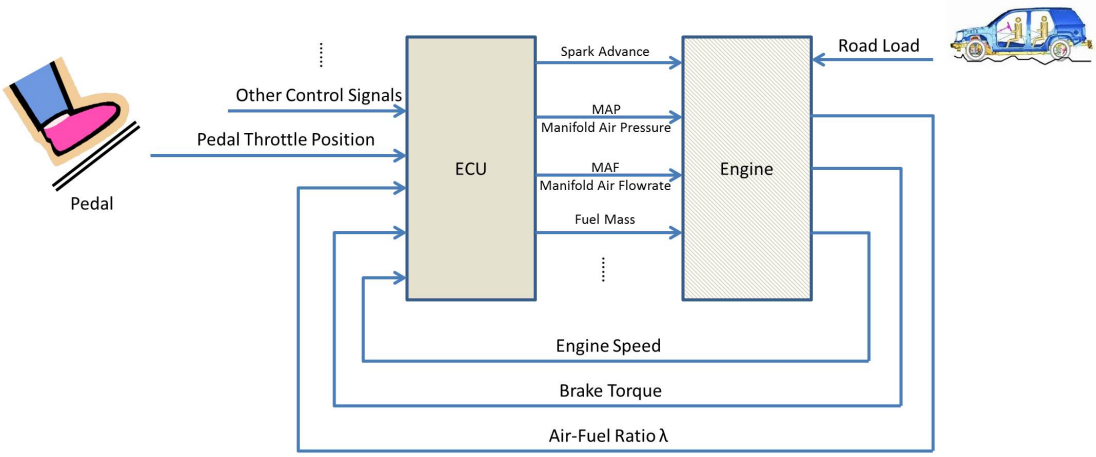
The power and torque generated by the IC engine is determined by many factors such as the combination of the fuel and air, the ignition time, and the propogation of the flame in the combustion chamber. The general engine control scheme is shown in Figure 1.12a. In the control perspective, the engine control unit (ECU) collects the data from various sensors that monitor the conditions of the intake air-fuel combination (e.g. fuel injection time, manifold air pressure MAP) , spark time, and the exhaust gas properties, computes the system variables of the related optimal combustion condition, transmits the control commands to the corresponding acuators (e.g. the throttle position, ignition command), and communicates the signals of exhaust gas property, engine speed, the brake torque, and other signals with other control units. The control structure of the ECU is complicated in reality. Nevetheless, to calibrate the ECU, the engine control system is, in general, modelled as a closed-loop control system such as the block

diagram of the system in Figure 1.12b.

In this section, four important control loops in the ECU are introduced in more detail : Air-Fuel Ratio Control, Ignition Control, Idle Speed Control, and Knock Control. In Chapter 6 of the thesis, to investigate the performances of the controllers by using different algorithms, one of the control loops, which determines the optimal spark advance (SA) in terms of the peak pressure positions (PPP) of the cylinders in order to generate the maximum brake torque (MBT) [76, 75], is used as an example.




(a) General gasoline engine control scheme [22]




(b) Block diagram of the engine control system

Figure 1.12: Modelling of the engine control system

### 1.2.1 Air-Fuel Ratio Control

Air-fuel ratio control is one of the most important tasks in automotive engine control. It is known [59] that, for a Spark-Ignition (SI) engine, the stoichiometric mixture of the air and fuel for the combustion is typically 14.7:1. The index $\lambda$ defined as the value of the air/fuel ratio over 14.7 determines the efficiency of the Three-Way-Catalytic (TWC) converter in removing hydrocarbon (HC), carbon monoxide (CO), and nitrogen oxide (NOx) emissions ( i.e. the system meets the emission regulations if $\lambda \sim 1$ ) as well as the fuel economy and power generation. With even small (5%) devidents, there exist significant pollutants, such as HC, CO, and NOx, in the exhaust gases. As a result, a good control strategy is required to determine the angle of electronic throttle to fix the manifold absolute pressure (MAP), and mass air flow (MAF). In addition, the opening and closing times of the fuel injector is also the key to achieving good emission control. As a result, most modern vehicles are now equipped with the universal exhaust gas oxygen sensors (UEGO) located in the air-fuel control path. The UEGO device allows feedback control of $\lambda$ and so greatly improves the performance of $\lambda$ control. Furthermore, TWC converters are commonly installed to reduce of the pollutants by chemical deoxidation. Nevertheless, there exists a problem of the reduction of the TWC converter's efficiency below the working temperature, which results in a high level of exhaust pollution. In summary, with the stringent regulations on vehicle emissions, the role of the air-fuel ratio controller in the ECU is an essential but a challenging problem for the automotive control engineers.

### 1.2.2 Ignition Control

The mixture of air and fuel in the engine combustion chamber requires to be ignited for generating the power. The control of the combustion is based on the timing of the ignition and the metering of the air and fuel. The ignition timing has a signitficant effect on the efficiency of the energy from the fuel. Conventionally, for a spark-ignition engine, the ideal ignition time is set before the crank reaches the top dead centre (TDC). The angle before TDC is known as the Spark Advance (SA). It is found in [51] that the peak pressure position (PPP) of the crank is highly related to the optimal ignition timing. As a result, the peak pressure position signal can be used to determine the optimal settings for the spark advance. In the typical on-road vehicle, the ignition controller computes the nominal ignition time in terms of the engine speed and other measured signals, such as the manifold pressure and the knock detector [35].

### 1.2.3 Idle Speed Control

In powertrain control systems, the idle speed controller is an important component in the engine management system (EMS). A typical vehicle is known to consume about 30 percent of the fuel at idle in cities [55]. Hence, the control of the idle speed significantly influences the overall fuel economy. It is estimated that reducing 100 rpm at engine idle speed theoretically improves the fuel efficiency for one mile per gallon in the standard Constant Volume Sample calculation [54]. However, engine stall limits the achievable fuel economy by increasing the neccessary engine speed in idle speed conditions. Typically, the range of idle speed in most engines is about 600 rpm to 1,000 rpm. In the idle speed control loop, the amount of air and fuel is respectively determined by the opening of the electronic throttle valve and the fuel injection time. To maintain the engine speed at idle, the optimal ignition time is often calculated in terms of the maximum best torque (MBT).

### 1.2.4 Knock Control

Knock phenomenon results from a self-ignition behaviour that produces very high pressure and locally high temperatures in the engine. Knock sometimes leads to catastrophic consequences, such as the damage of the cylinder and the rim of the piston. As a result, it is important to adjust the control parameters to prevent knock when it occurs. The knock control strategy is usually to retard the spark advance or decrease the boost pressure of the turbocharger until the knock phenomenon disappears and then to gradually advance the SA and increase the boost pressure thereafter [82].

## 1.3 Overview of the Thesis

This thesis presents an analysis of the different analytic function methods used to solve nonparametric optimization problems in the $\mathbb{H}^\infty$ method. The summary of each chapter is outlined below.

**Chapter 2** The chapter presents the review of the Disk Iteration (DI) method proposed by Helton and Merino [46] and its implementation. The optimization problem in the DI method is reviewed in Section 2.1 where the approximation to the original $\mathbb{H}^\infty$ optimization problem is summarized. The two nonparametric spectral factorization methods in the DI method are studied in the beginning of Section 2.2. The solution to Nehari problem related to the original optimization

problem by using the Nehari-commutant-lifting formula is also discussed in this section. The algorithm of the DI method and the implementation of the DI method are presented in the latter part of Section 2.2.

**Chapter 3** This chapter re-states the theory of the Newton Iteration (NI) method proposed by Helton et al. [39] and proposes the Matlab implementation of the NI method. In Section 3.1, the optimality conditions of the $\mathbb{H}^\infty$ optimization problem are presented. The operator equation in the NI method is stated in the start of Section 3.2 where the solution to the operator equation is re-produced from the results in [39]. In terms of the Newton iteration procedure, the implementation of the Jacobian operator approaches the optimal analytic solution to the $\mathbb{H}^\infty$ optimization problem. Later in Section 3.2, the matrix implementation of the two components as the Conjugate Toeplitz operator and the Shifted Hankel operator is presented. The update increment function is found by the inverse of the Jacobian operator in the NI method at the end of this chapter.

**Chapter 4** The chapter shows three numerical examples to analyze the above two analytic function methods. The formulation of each example is described in the first three sections of this chapter. The discussion of the DI and NI methods in terms of their convergence rate is addressed in Section 4.4. The finding of the NI method's property in the three examples is discussed in Section 4.5.

**Chapter 5** This chapter studied a nonparametric method proposed by Helton and Sideris [40] in terms of the Hadamard $\mathbb{H}^\infty$- Frobenius norm in the mixed sensitivity control. The nonparametric methods for the internal stability requirement are discussed at the begining of Section 5.2 where the Hadamard $\mathbb{H}^\infty$- Frobenius norm problem proposed by Diggelen and Glover [92] is also reformed for nonparametric control. The algorithm of the LP method is given in the same section. The linear programming optimization problem using Streit's algorithm [86] is reviewed in Section 5.3. An example problem due to Skogestad and Postlethwaite [83] is used to investigate the nonparametric LP method in Section 5.4.

**Chapter 6** This chapter presents an application of the nonparametric $\mathbb{H}^\infty$ controller design method on an automotive Pressure-Peak-Position control problem. In Section 6.1, the formulation of the problem is addressed. The single sensitivity control problem is addressed using the proposed analytic function methods, i.e. the DI, NI, and LP methods, and presented in Section 6.2. In Section 6.3, the mixed sensitivity control problem is considered by means of the three ana-

lytic function methods. The analysis and comparison of these analytic function methods are summarized in Section 6.4.

**Chapter 7** This chapter outlines the results presented in the previous chapters and proposes the future research direction of the work in this thesis.

## 1.4 Contributions of the Thesis

The claims to novelty in the thesis are

- The Matlab implementation of the DI method originally proposed by Helton et al. [46] is presented for the purpose of comparison and analysis with other algorithms. The nonparametric spectral factorization methods by Wilson [99] and Harris and Davis [36] are also coded in Matlab and incorporated as part of the optimization code. Although the solution of the DI method is not optimal to the general $\mathbb{H}^\infty$ optimization problem, the optimization problem can be approached by iterating to the closest circular form and solved in terms of the Nehari problem. Thus, its sup-optimal solution is still useful for nonparametric control and, in the problems with circular performance function, an optimal solution is obtained. Since the original programs that implement the DI method were coded in Mathematica [41], for use in engineering applications, it is practically useful to integrate a Matlab version of the DI method as part of a controller design toolbox.

- An alternative nonparametric method known as Q-parameterization or Youla-Kucera parameterization of sensitivity functions by the sensitivity function $Q$ can be used in order to meet the internal stability conditions in the feedback $\mathbb{H}^\infty$ problem and to replace the parametric interpolation method proposed by Helton et al. [47, 106]. The interpolation method with the DI method has previously been implemented on an automotive engine control problem by using the Frobenius $\mathbb{H}^\infty$ norm [108], however, this interpolation method required the parametric identification of the RHP pole and zero of the plant. The Q-parameterization method skips this parametric identification process and thereby allows the approach to become fully nonparametric method. In this thesis, the Q-parameterization method is successfully applied to an automotive engine control problem with nonparametric optimization algorithms.

- The implementation of the NI method originally proposed by Helton et al.[39] is made and its effectness is investigated. Compared to the DI

method, the NI method is capable of dealing with multiple function optimization with non-circular type objective functions. The solution by the NI method is potentially optimal to the general optimization problem. Furthermore, the NI method should in principle converge quickly to the optimal solution due to its second order convergence rate. These properties of the NI method potentially save cost in computing effort during the optimization process as well as improving the accuracy of the solution. The NI method is therefore more suitable for general nonparametric control problems. The NI method is also implemented in Matlab in this thesis for the purpose of analysis and practical use. This implementation is the first published implementation of this important algorithm.

- The comparative analysis of the DI method and the NI method is presented. This is the first independent comparison of these two methods. The analytic solutions by both methods are investigated by applying to several numerical examples.

- A nonparametric approach by Streit's [86] linear programming (LP) method for $\mathbb{H}^\infty$ control problems is proposed. Compared to that the DI method and the NI method which are both based on the computation of the Fourier coefficients of frequency response function, the LP method is formed by the approximation of the analytic solution directly on the basis of the frequency response itself. The central optimization approach originally studied by Helton and Sideris [40] using Streit's method [86] is described in this thesis. The implementation of Streit's algorithm is also coded in Matlab for practical use and for comparison purpose. An example of a SISO control problem with multiplicative uncertainty by this nonparametric approach is demonstrated.

- The nonparametric approach using the different analytic function methods, i.e. the DI method, the NI method and the LP method, is applied to an automotive engine control problem. The analysis of the different controllers from the different analytic function methods is presented.

# Chapter 2

# Disk Iteration Method for $\mathbb{H}^\infty$ Optimization

The standard $\mathbb{H}^\infty$ control problem is often viewed as a mathematical optimization problem. The optimization problem mentioned in Chapter 1 can be solved by the Disk Iteration (DI) method proposed by Helton and Merino [46, 47]. Most parts of this chapter are originated from the work of Helton and Merino[46, 47] and the programs to implement the DI method are also available in [41]. However, the Matlab implementation of the DI method is firstly accomplished for the purpose of comparison and practical use.

In this chapter, starting with the statement of the optimization problem and the transformation from the $OPTe$ problem to the $OPTd$ problem in Section 2.1, the underlying theory and the implementation of the Disk Iteration (DI) algorithm are presented in more details in Section 2.2.

## 2.1 Optimization Problem

This section is the summary of the work on $\mathbb{H}^\infty$ optimization in the frequency domain by Helton et al in [37, 47] and in other related publications [38, 39, 40, 42, 43, 45, 46, 48]. In this section, the $\mathbb{H}^\infty$ control problem in the frequency domain approach is formulated by using Equation 1.24 . Equation 1.24 specifies a mathematical minimax problem :

*Given a continuous positive-valued function* $\Gamma$ *in* $\mathbb{C}^N$ *where* $N$ *is the dimension of the problem, and a set of all the stable functions on the* $j\omega$ *axis ( denotes as* $\mathbb{RH}^\infty$ *, the space of all functions whose poles have negative parts. ), find the optimal function* $f^*(j\omega) \in \mathbb{RH}^\infty$ *such that*

$$\gamma = \inf_{f \in \mathbb{RH}^\infty} \sup_{\omega} \Gamma(\omega, f(j\omega)) \tag{2.1}$$

*where* $\omega$ *represents the frequency, and* $f(j\omega)$ *is the frequency response function*

This problem is known as the $OPT$ problem in [48]. It can be translated into an optimization problem that searches the stabilizing controller in terms of the measure $\gamma$ in the worst performance case.

In mathematical analysis, the functions are often treated on the unit disk rather than in the complex plane because of the use of the Fourier Transform, the ease of implementation in the computer codes and other properties in complex analysis. The transformation from the imaginary axis to the unit circle $\partial D$ is possible via a Linear Fractional Transform (LFT). A LFT is a one-to-one, linear, and bounded mapping in this case from the imaginary axis ( $s = j\omega$ ) onto the unit circle $\partial D$. It transforms the poles and zeros of the function in the left half plane onto a point inside the unit cirle and maps the poles and zeros of the function in the right half plane onto the point outside the unit circle. The sampling points on the imaginary axis ( e.g.. $j\omega$ ) are thus mapped to points on the unit circle $\partial D$ by the linear fractional transform, as illustrated in Figure 2.1.

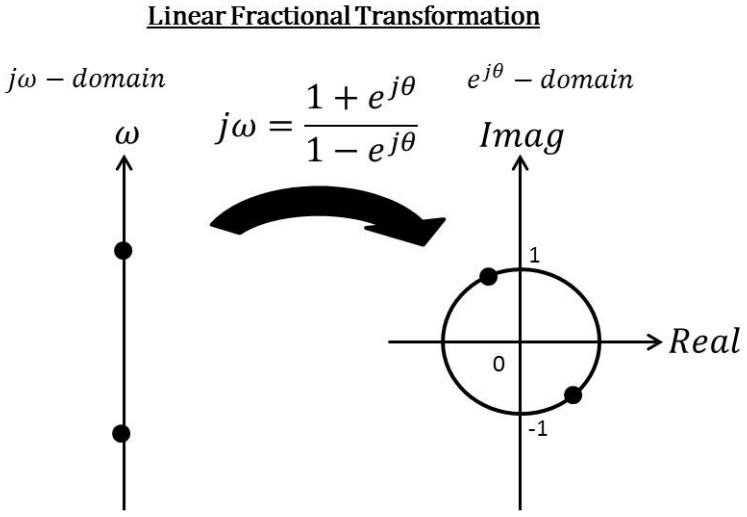



Figure 2.1: Linear fractional transform

Thus by means of the linear fractional transform, the function $f(j\omega)$ in $\mathbb{RH}^\infty$ in the $OPT$ problem is mapped to the function $f\left(e^{j\theta}\right)$ in $\mathbb{A}$, which is the subspace

of all analytic functions $f\left(e^{j\theta}\right)$ that are bounded and continuous in the unit disk $D$. The $OPT$ problem is therefore transformed to the so-called $OPTe$ problem given by

*OPTe* *Given a continuous, real, positive valued function* $\Gamma$ *in* $D$*, find the optimal function* $f^*\left(e^{j\theta}\right)$ *such that*

$$\gamma^* = \inf_{f^*\in\mathbb{A}} \sup_{\omega} \Gamma\left(e^{j\theta}, f^*\left(e^{j\theta}\right)\right) = \inf_{f^*\in\mathbb{A}} \left\|\Gamma\left(e^{j\theta}, f^*\left(e^{j\theta}\right)\right)\right\|_\infty \tag{2.2}$$

*where* $e^{j\theta}$ *are the points spaced around the unit circle,* $f\left(e^{j\theta}\right)$ *is continuous in* $\mathbb{H}^\infty$ *and* $*$ *denotes the function value at the optimum.*

For simplicity, we will denote $z = e^{j\theta}$ in the rest of the thesis. It is stated in [45] that the $OPTe$ problem is closely related to the sub-optimization problem $OPTc$ problem in terms of the sublevel set $S_\theta(c) = \{\Gamma(z, f(z)) \leq c, c \in \mathbb{R}^+\}$with a prescribed performance function $c \in \mathbb{R}^+$ :

*OPTc* *Find* $f^*(z) \in \mathbb{A}$ *with* $\Gamma(z, f(z)) \leq c$*,* $\forall\theta$*, and* $c \in \mathbb{R}^+$*such that*

$$\gamma^* = \inf_{f^*\in\mathbb{A}} \left\|\Gamma\left(z, f^*(z)\right)\right\|_\infty = \inf_{f^*\in\mathbb{A}} \left\|k(z) - f^*(z)\right\|_\infty^2 \leq c \tag{2.3}$$

*where* $k(z)$ *is any complex-valued function*

It is also shown in [45] that the sublevel sets $S_\theta$ have the shape of disk and their properties of boundness, simple connectness, smoothness and convexity are closely related to the properties of solutions to the $OPTe$ problem. Therefore, the solution to the $OPTc$ problem can be viewed as the solution to the $OPT$ problem.

Furthermore, in [46, 47], the generalization of the above $OPTc$ problem for $N \geq 1$ is considered :

*OPTd* *Given the positive continuous functions* $w(z)$ *and* $k(z)$*, find* $f(z) \in \mathbb{H}_N^\infty$ *such that*

$$\gamma = \inf_{f\in\mathbb{H}_N^\infty} \sup_{\theta} \Gamma(z, f(z)) = \inf_{f\in\mathbb{H}^\infty} \sup_{\theta} \sum_{i=1}^{N} w_j(z) \left|k_i(z) - f_i(z)\right|^2 \tag{2.4}$$

Two examples to illustrate the relationship between the $OPT_d$ and the $OPTc$ problems are discussed in the following.

*Example1*[47] Given a scalar-valued function ( i.e. the $OPTd$ problem for $N = 1$ )

$$\begin{aligned} \Gamma(z,f) &= w(z)\left|k(z)-f(z)\right|^2 && (2.5)\\ &= (2+\cos\theta)\left|\frac{1}{z+0.5}-f\right|^2 && (2.6)\\ &\leq \gamma \end{aligned}$$

it is seen that the solution must lie in the disk centred at $\frac{1}{z+0.5}$ with the radius $\sqrt{\gamma/(2+\cos\theta)}$. This can be related to the $OPTc$ problem by observing that the sublevel sets of the solutions $f$ are

$$f = \left\{ f \in \mathbb{H}^\infty; \left|k-f\right|^2 \leq r^2, \text{for } r \in \mathbb{R}^+ \right\}$$

where the centre function $k = \frac{1}{z+0.5}$ and the radius function $r = \sqrt{\gamma/(2+\cos\theta)}$. Consequently, the equivalent $OPTc$ problem for this objective function $\Gamma(z,f)$ can be reformed as

$$\begin{aligned} \inf_{f\in\mathbb{H}^\infty}\sup_\theta \Gamma(z,f) &= \inf_{f\in\mathbb{H}^\infty}\sup_\theta (2+\cos\theta)\left|\frac{1}{z+0.5}-f\right|^2 && (2.7)\\ &= \inf_{f\in\mathbb{H}^\infty}\sup_\theta \left|k-\tilde{f}\right|^2 && (2.8) \end{aligned}$$

where $k = \left(\sqrt{\gamma/(2+\cos\theta)}\right)^{-1/2}\cdot\frac{1}{z+0.5}$ and $\tilde{f} = \left(\sqrt{\gamma/(2+\cos\theta)}\right)^{-1/2} f$

As the $OPTc$ problem is solvable by solving the Nehari problem, we immediately have the solution to the optimization for this type of $\Gamma(z,f)$.

*Example2*[47] An objective function that has more than one term of the form :

$$\Gamma(z,f) = w_1\left|f\right|^2 + w_2\left|1-f\right|^2 \leq \gamma \qquad (2.9)$$

it can be considered a *quasi-circular* problem. The sublevel sets of the function $\Gamma(z,f)$ are fomulated as the disks centred at $\frac{w_2}{w_1+w_2}$ with the radius $\left(\frac{1}{(w_1+w_2)}\left(\gamma-\frac{w_1w_2}{w_1+w_2}\right)\right)^{1/2}$. Furthermore, the problem can be written in terms of

$$f = \left\{ f \in \mathbb{H}^\infty; \left|\frac{w_2}{w_1+w_2}-f\right|^2 \leq \frac{1}{(w_1+w_2)}\left(\gamma-\frac{w_1w_2}{w_1+w_2}\right)\right\}$$

This implies that the problem can be viewed as a *circular* problem with the centre function $\frac{w_2}{w_1+w_2}$ and the radius function $\left(\frac{1}{(w_1+w_2)}\left(\gamma-\frac{w_1w_2}{w_1+w_2}\right)\right)^{1/2}$. There-

fore, it is concluded that any function in the form of

$$\Gamma(z, f) = \sum_{i=1}^{N} w_i(z) |k_i(z) - f_i(z)|^2$$

is said to be a *quasi-circular* function, which is solvable by means of its equivalent circular form.

In summary, whether the function is circular or quasi-circular, the $OPTd$ problem can be solved by solving the equivalent $OPTc$ problem in terms of the solution to the Nehari problem. To deal with other functions that are not of circular or quasi-circular form, the approximation to the nearest circular function is performed. In the ANOPT package [41], the approximation is coded in terms of using an interpolation method. The interpolation method is discussed in depth in Section 2.2.

## 2.2 Disk Iteration Method

This section describes the *Disk Iteration* method that is presented by Helton and Merino in [46]. The *Disk Iteration* method was implemented for application to several control problems [106, 108] in a very computationally effective way. Computer code for the *Disk Iteration* method is available from the open source program ANOPT package in Mathematica [41].

It is valueable to translate the ANOPT programs into the Matlab language. Since most control engineering work is done in Matlab, a procedural version of the ANOPT package was currently not available before the work in this thesis.

This section outlines the underlying algorithm in the ANOPT package. Starting the $OPTd$ problem in the previous Section 2.1 and the investigation of the spectral factorization methods in Section 2.2.1, the solution to the $OPTe$ problem is attainable by solving the Nehari problem in terms of the commutant lifting theorem. The description of the solution to the Nehari problem is presented in Section 2.2.2. The DI algorithm is then described in Section 2.2.3. The implementation of the algorithm is described in details in Section 2.2.4.

### 2.2.1 Spectral Factorization

From the two examples in the previous Section 2.1, it is seen in Equation 2.10

$$\begin{aligned}\Gamma_i(z,f) &= w_i(z)\left|k_i(z)-f_i(z)\right|^2 = \sigma(z)\,\sigma^*(z)\left|k_i(z)-f_i(z)\right|^2 \\ &= \left|\sigma_i(z)\right|^2\left|k_i(z)-f_i(z)\right|^2 = \left|\tilde{k}_i(z)-\tilde{f}_i(z)\right|^2 \end{aligned} \tag{2.10}$$

for $i=1,2,3,\cdots,N$ where $\tilde{k}_i(z)$ and $\tilde{f}_i(z)$ are also in $\mathbb{H}^\infty$ and $\tilde{k}_i(z)=k_i(z)\cdot\sigma(z)$ , $\tilde{f}_i(z)=f_j(z)\cdot\sigma(z)$ and $w_i(z)=\sigma(z)\,\sigma^*(z)$ where $\sigma(z)$ is known as the spectral factor of $w_i(z)$ whereas $\sigma^*(z)$ denotes the complex conjugate of $\sigma(z)$ that the function $w_i$ outside the absolute value bracket is moved into the bracket and the objective function is re-formulated to the equivalent square form. It is important to keep the analytic property of the solution function $\tilde{f}_i$ remain unchanged when multiplying the function $w_i^{1/2}$. The method to factor $w_i$ is generally known as 'spectral factorization' method and specifically used in Equation 2.10.

Spectral factorization methods have been studied for many years and are still an important topic in filtering problems and linear control problems since Wiener [95] worked on factoring the scalar spectrum of a random sequence as well as the extension to matricial spectral factorization problems [96, 97]. After Youla [100] developed a method to the spectral factorization problem e.g. $w_i(z)=\sigma(z)\,\sigma^*(z)$ both parametric approaches and non-parametric approaches have been actively investigated in signal processing and optimal control [78]. An overview of parametric methods can be found in [61]. In this section, two non-parametric approaches by Wilson [98] and Harris [36] are presented below.

#### 2.2.1.1 Wilson's method [98]

The method was presented by Wilson [98] for the scalar problems. Wilson's spectral factorization method was derived on the basis of Bauer method [10, 11], which is to approximate the coefficients of the spectral factor by computing the Cholesky decomposition of the associated Toeplitz matrix. More specifically, given a matrix-valued function $f\in\mathbb{C}^{N\times N}$ defined almost everywhere in the interval $[-\pi,\pi]$, if $f(\theta)$ is Hermitian and nonnegative and $f(-\theta)=f(\theta)^T$, we can map the function onto the unit circle in terms of the Fourier series as

$$f(\theta)=\sum_{k=-\infty}^{\infty}\gamma_k\cdot e^{jk\theta}=\sum_{k=-\infty}^{\infty}\gamma_k\cdot z^k$$

where $\gamma_k$ are the Fourier coefficients given by $\gamma_k=(1/2\pi)\int_{-\pi}^{\pi}f(\theta)\,e^{-jk\theta}d\theta$.

It is known [99] that the following conditions hold if $f(\theta)$ is defined as above

1. $f(\theta)$ has full rank almost everywhere (denoted as *a.e.* afterwards)
2. $\int_{-\pi}^{\pi} \log\det f(\theta)\, d\theta > -\infty$
3. $f(\theta) = \sigma(z)\,\sigma^*(z)$ where $\sigma(z)$ is the generating function which is in the $N \times N$ matrix form defined on the unit circle with the properties

   (a) $\sigma(z)$ has Fourier series expansion where all the negative coefficients vanish :

   $$\sigma(z) = \sum_{k=0}^{\infty} A_k z^k$$

   where

   $$A_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} \sigma(z)\, e^{-jk\theta} d\theta$$

   (b) $\sigma(z)$ is analytic in the unit disk, i.e. $\left\{\sigma(z) = \sum_{k=0}^{\infty} A_k z^k, |z| < 1\right\}$

   (c) $\sigma(z)$ is optimal, i.e. $\det\sigma(z) \neq 0$

Taking the logarithm on both sides of the above *Condition* 3, we have

$$\log f(\theta) = \log\sigma(z) + \log\sigma(z)^*$$

Because the functions $f$ and $\alpha$ can be expanded by Fourier series as

$$\begin{aligned} \log f(\theta) &= \sum_{k=-\infty}^{\infty} \alpha_k z^k \\ \log\sigma(z) &= \sum_{k=0}^{\infty} \beta_k z^k \end{aligned}$$

this leads to

$$\log\sigma(z) = [\log f(\theta)]_W^+$$

i.e.

$$\sigma(z) = \exp[\log f(\theta)]_W^+$$

where $[\cdot]_W^+$ is defined as

*Given* $g(\theta) = \sum_{k=-\infty}^{\infty} G_k z^k$,

$$[g(\theta)]_W^+ = \frac{1}{2}G_0 + \sum_{k=1}^{\infty} G_k z^k \tag{2.11}$$

For $N = 1$, the computation of the spectral factor therefore becomes very easy, which is coded in a procedural language in the following steps :

1. *Take the logarithm of* $f(\theta)$
2. *Calculate the Fourier coefficients of the function* $\log f(\theta)$ *in Step 1 by using fast Fourier transform (FFT)*
3. *Compute the function* $[\log f(\theta)]^+$ *by applying the inverse fast Fourier transform (IFFT) to the coefficient vector* $\begin{bmatrix} \cdots & 0 & \frac{\beta_0}{2} & \beta_1 & \beta_2 & \cdots \end{bmatrix}$ *calculated in Step 2.*
4. *The spectral factor is then obtained :* $\sigma = \exp[\log f(\theta)]_W^+$

Wilson's spectral factorization method for a scalar function $f(\theta)$ is very effective and reliable. However, as reported in [99], this is not valid when $f(\theta)$ is a matrix-valued function because there is no computable extension of logarithms to matrices. As a result, Wilson made modifications in [99] to linearize the problem and presented an iterative solution in terms of Newton-Raphson method. Therefore, the Wilson's method is also able to deal with matrix-valued functions. Nevetheless, in the DI method, the matrix-valued function $\Gamma(z, f(z))$ in the $\mathbb{H}^\infty$ optimization problem is often seperated in individual domain whose the disk-shaped properties provide the existence of analytic solutions. The details will be discussed in Section 2.2.2. Thus the matrix version of the spectral factorization is not required in the DI method. The scalar Wilson's method is therefore considered an effective algorithm for NP spectral factorization.

#### 2.2.1.2 Harris and Davis' method [36]

Another nonparametric spectral factorization method was presented by Harris and Davis[36]. Suppose we are given an $N \times N$ matrix-valued function $H(z)$ defined on the unit circle with the following properties in [36]

1. $H(z)$ is Hermitian, i.e. $H(z) = H(\bar{z})^T$
2. $H(z)$ is positive definite, i.e. $H(z) > 0$
3. $H(z)$ is bounded and can be approximated by the Fourier series, i.e.

$$H(z) = \sum_{k=-\infty}^{\infty} H_k z^k = \sum_{k=-\infty}^{\infty} H_k e^{jk\theta}$$

It is then possible to derive from the result in [31] that

$$H(z) = \sigma(z) \cdot W \cdot \sigma^*(z)$$

where $W$ is a positive definite matrix and

$$\sigma(z) = \sum_{k=0}^{\infty} \sigma_k z^k = \sum_{k=0}^{\infty} \sigma_k e^{jk\theta}$$

with $\sigma_0 = I$.

The spectral factorization method is based on the results of [19], which is given by the iteration

$$\sigma_{n+1}(z) = W_n^{-1} \left[\sigma_n^*(z)^{-1} H(z) \sigma_n(z)^{-1}\right]_H^+ \sigma_n(z)$$

$$\text{for } z = e^{j\theta} \text{ and } \theta \in [0, 2\pi]$$

where the projection $[\cdot]^+$ in this spectral factorization method of a function of the form $G(z) = \sigma_n^*(z)^{-1} H(z) \sigma_n(z)^{-1} = \sum_{k=-\infty}^{\infty} G_k z^k$ is defined as

$$[G(z)]_H^+ = \sum_{k=0}^{\infty} G_k z^k$$

and $W_n$ is the mean value of the function $\sigma_n^*(z)^{-1} H(z) \sigma_n(z)^{-1}$.

Furthermore, the projection $[\cdot]_H^+$ can be calculated by the formula [50] : $[G(z)]_H^+ = \frac{1}{2}[W_n + G + j\mathbb{H}[G]]$ where $\mathbb{H}[G]$ is the Hilbert transform of the function $G(z)$. This allows the use of the standard Matlab function `hilbert` in the program to implement Harris' method.

#### 2.2.1.3 Comparison

Both the Wilson and Harris-Davis methods are very effective for nonparametrically factoring a scalar function. However, the Harris-Davis method requires more computation efforts due to the implicit iteration routine. On the other hand, the Wilson's method only needs the computation of the logarithm. Therefore, it is expected that the Wilson's method is faster than the Harris-Davis method. A simple scalar-valued example is shown below to compare the two methods. The two methods also work for matrix-valued functions. However, as previously discussed, in the DI method, only scalar spectral factorization technique is required.

Given the frequency response of a scalar-valued function $R(z) = \left|\frac{z+5}{(z+1)^2}\right|$, we want to find the spectral factor $\sigma$ as $R = \sigma \cdot \sigma^*$. The frequency response is calcu-

lated by choosing the sampling points $z_k = 1/\tan(k/N)$ for $k = 1, 2, 3, \cdots, N-1$ where $N$ is the number of sampling points. A comparison of the results of the two methods is shown in Figure 2.2 and Table 2.2.1. It is observed that both methods converge to the same solution (as shown in Figure 2.2) with the numerical errors 4.5040E-8 and 2.945E-8 respectively. Nevertheless, the computation by Wilson's method took 0.0022 seconds and the computing time was 0.0291 seconds for Harris-Davis' method, which meets our expectation. As a result, in the Matlab programs developed in this thesis for scalar applications, the spectral factorization implements Wilson's method as the main solver and the Harris-Davis' method as the alternative in case of the situation that Wilson's method fails to converge within the tolerance.

| Algorithm | Computing Time | Errors $\max \lvert R - \sigma\sigma^* \rvert$ |
|---|---|---|
| Wilson | 0.0022 sec | 4.5040E-8 |
| Harris | 0.0291 sec | 2.2945E-8 |

Table 2.2.1: Comparison of Wilson and Harris-Davis methods

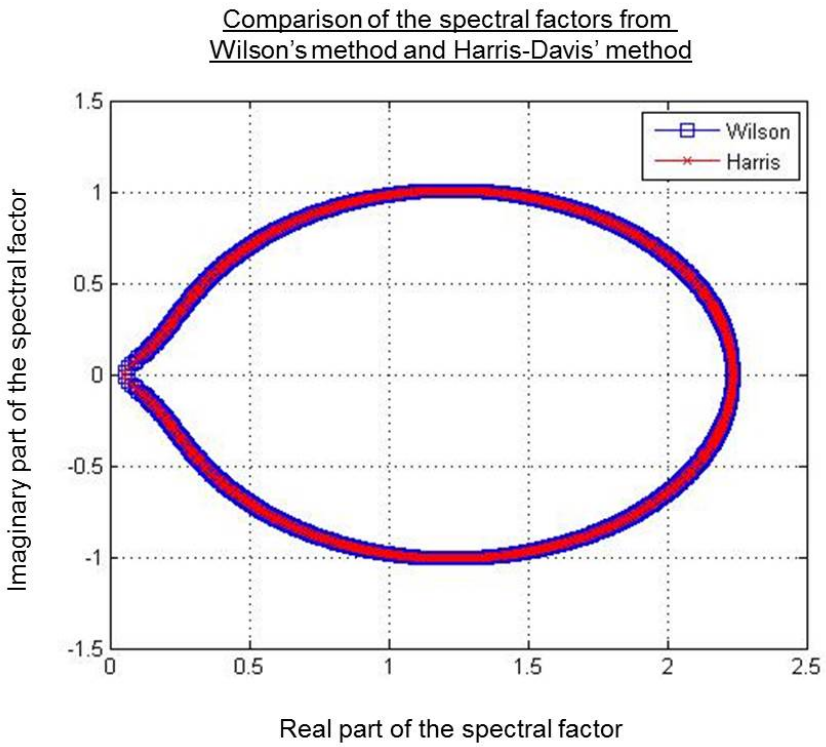



Figure 2.2: Comparison of the spectral factors from Wilson's and Harris-Davis' methods

### 2.2.2 The Nehari Problem and Commutant Lifting Theorem

In this section, the solution to the $OPTc$ problem is presented. Most of the contents in this section is from the collection of the work in [39, 47]. It is observed that the $OPTc$ problem can be solved by solving Nehari problem [39] :

***Nehari*** *Given a function $k \in \mathcal{L}_N^\infty$, there exists a function $f^* \in \mathbb{H}_N^\infty$ such that the distance between $k$ to $\mathbb{H}^\infty$ is*

$$dist\left(k, \mathbb{H}^\infty\right) = \inf_{f \in \mathbb{H}^\infty} \|k - f\|_{\mathcal{L}^\infty}$$

It is observed that Nehari problem is in the same form of the $OPTc$ problem if

$$\Gamma\left(z, f\right) = \|k\left(z\right) - f\|_N^2 \tag{2.12}$$

It is also known that the solution to the Nehari problem is determined by the Hankel operator $\mathcal{H} : \mathbb{H}^2 \to \mathbb{H}^{2\perp}$ with the action $\mathcal{H}\left[f\right] = P_{\mathbb{H}^{2\perp}}\left[kf\right]$ where $f, k \in \mathcal{L}_\infty^N$ and the operator $P_{\mathbb{H}^{2\perp}}\left[\cdot\right]$ returns the function's Fourier coefficients whose indexes are negative. The norm of the Hankel operator $\mathcal{H}$, which is constructed from the negative Fourier coefficients of $K$, equals the minimal distance as its infimum is attained. In other words, if there is a maximizing vector $\alpha$ such that $\|\mathcal{H}_\alpha\| = \|\mathcal{H}\| \|\alpha\|$, the theory by Adamjan, Arov, and Krein[3, 4, 5, 77] gives that the unique best $\mathbb{H}^\infty$ approximation $f^*$ to $k$ is

$$f^*\left(z\right) = k\left(z\right) - \frac{\mathcal{H}_\alpha}{\alpha} \tag{2.13}$$

In other words, Equation 2.13 can be arranged as

$$k\left(z\right) - f^*\left(z\right) = \frac{\mathcal{H}\left[\alpha\right]}{\alpha} \tag{2.14}$$

i.e.

$$\mathcal{H}\left[\alpha\right] = \alpha\left(k\left(z\right) - f^*\left(z\right)\right) \tag{2.15}$$

Suppose the adjoint of the operator $\mathcal{H}$ is denoted as $\mathcal{H}^*$ then

$$\mathcal{H}^*\left[\alpha\right] = \alpha\overline{\left(k\left(z\right) - f^*\left(z\right)\right)}$$

and we have

$$\mathcal{H}^*\left[\mathcal{H}\left[\alpha\right]\right] = \gamma^2\alpha \tag{2.16}$$

where $\gamma = \inf_{f\in\mathbb{H}^\infty} \|k - f\|_{\mathcal{L}^\infty}$ as defined in Problem $OPTe$ ( i.e. Equation 2.2 ).

This is generally known as the *Nehari-commutant lifting formula* [3, 4, 5, 69, 77], which provides the solution to the OPTc problem. More details of the derivations can be found in [46, 47, 49].

### 2.2.3 The Algorithm of the Disk Iteration Method

The section is the summary of the results in [39]. The algorithm for solving the $OPTd$ problem is presented in this section. Given a small perturbation function $h \in \mathbb{H}^\infty$, expand $\Gamma\left(z, f+h\right)$ in Taylor series about $f$ up to the second order, i.e.

$$\begin{aligned}\Gamma\left(z, f+h\right) &= \Gamma\left(z, f\right) + 2\Re\left(\frac{\partial}{\partial z}\Gamma\left(z, f\right)^T h\right) &(2.17)\\ &\quad + \bar{h}^T \frac{\partial^2}{\partial\bar{z}\partial z}\Gamma\left(z, f\right) h + \Re\left(h^T \frac{\partial^2}{\partial z^2}\Gamma\left(z, f\right) h\right) + \mathcal{O}^3 &(2.18)\end{aligned}$$

the $OPTd$ problem becomes

*Given a continuous positive-valued function* $\Gamma$ *in* $\partial D$ ,

*the subspace* $\mathbb{A}$ *of the analytic functions on the unit circle, and a small function* $h \in \mathbb{H}^\infty$, *find the optimal function* $f^*\left(e^{j\theta}\right)$ *such that*

$$\begin{aligned}\gamma^* &= \inf_{f\in A}\sup_{\theta}\left\{\Gamma\left(z, f\right) + 2\Re\left(\frac{\partial}{\partial z}\Gamma\left(z, f\right)^T h\right)\right. &(2.19)\\ &\quad \left. + \bar{h}^T \frac{\partial^2}{\partial\bar{z}\partial z}\Gamma\left(z, f\right) h + \Re\left(h^T \frac{\partial^2}{\partial z^2}\Gamma\left(z, f\right) h\right) + \mathcal{O}^3\right\} &(2.20)\end{aligned}$$

*where* $z = e^{j\theta}$ *are the points spaced on the unit circle, and* $f^*\left(z\right)$ *is an analytic function in* $\mathbb{H}^\infty$.

In [39], the algorithm is developed in the sense of the coordinate descent method. The solution is iterated in terms of $f^{k+1} = f^k + th$ as described in the following :

***(i)*** *Find $h \in \mathbb{H}^\infty$ that minimizes*

$$\sup_{\theta} \Gamma(z, f) + 2\Re\left(\frac{\partial}{\partial z}\Gamma(z, f)^T h\right) + H_{h,h} \tag{2.21}$$

*where $H_{h,h}$ represents the collection of all the second order terms in Equation 2.19*

***(ii)*** *Find a positive real-valued function $t \geq 0$ such that*

$$\gamma = \inf_{f \in \mathbb{H}^\infty} \sup_{\theta} \|\Gamma(z, f + th)\|_{\mathbb{H}^\infty} \tag{2.22}$$

***(iii)*** *Update the current solution by $f^{k+1} = f^k + th$. Stop the iteration if the optimum is found*

Step (i) calculates the descent direction in terms of the increment function $h$ and Step (ii) gives the step length in the update process. The next solution is updated by adding the increment function $th$ in Step (iii). The illustration of the algorithm to update the solution is shown in Figure 2.3.

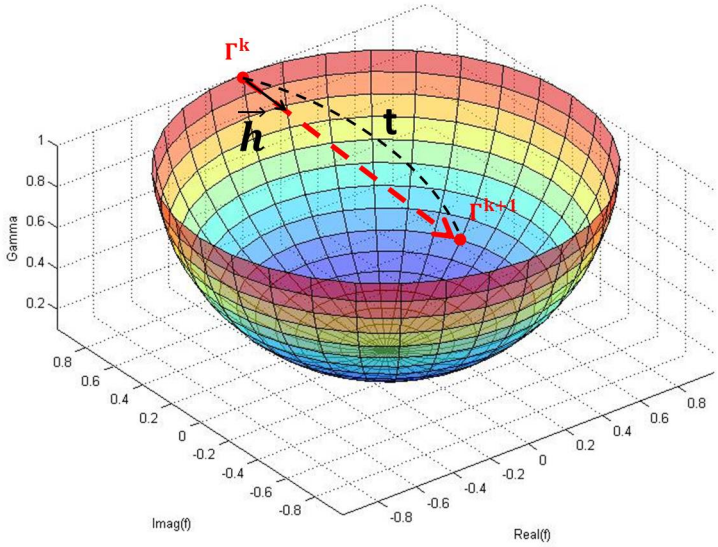



Figure 2.3: Disk Iteration method

In Equation 2.17, if the objective function $\Gamma(z, f)$ is of the circular form, it can be expressed as

$$\begin{aligned}\Gamma(z, f+h) &= g + 2\Re(ah) + H_{h,h} && (2.23)\\ &= g + 2\Re(ah) + h^T A h && (2.24)\\ &= g + 2\Re(ah) + w|h|^2 && (2.25)\end{aligned}$$

where $w$ is a real positive-valued function, $g = \Gamma(z, f(z))$, $a = \frac{\partial}{\partial z}\Gamma(z, f(z))$ and $A$ is a positive definite $N \times N$ function.

Furthermore, the function $\Gamma(z, f(z))$ can be reformulated as

$$\begin{aligned}\Gamma(z, f(z)) &\approx g + 2\Re(ah) + w|h|^2 + w\left|\frac{a}{w} + h\right|^2 && (2.26)\\ &\quad -w\left(\left|\frac{a}{w}\right|^2 + 2\Re\left(\frac{a}{w}h\right) + |h|^2\right) && (2.27)\\ &= g + 2\Re(ah) + w|h|^2 + w\left|\frac{a}{w} + h\right|^2 && (2.28)\\ &\quad -\frac{|a|^2}{w} - 2\Re(ah) - w|h|^2 && (2.29)\\ &= g - \frac{|a|^2}{w} + w\left|\frac{a}{w} + h\right|^2 && (2.30)\end{aligned}$$

A choice of the function $w$ as $w = \frac{|a|^2}{g}$ leads to $\Gamma(z, h) = w\left|\frac{a}{w} + h\right|^2 = \frac{|a|^2}{g} \cdot \left|\frac{g}{\bar{a}} + h\right|^2$ ($|a|^2 = a \cdot \bar{a}$). This is the typical $OPTc$ problem as discussed in Section 2.1 and we can immediately obtain the solution by using the Nehari-commutant lifting formula [47] to solve the related Nehari problem. Moreover, for other functions where $w \neq \frac{|a|^2}{g}$, we consider

$$\gamma^* = \inf_{f \in \mathbb{H}^\infty} \sup_\theta \Gamma(z, f) \approx \inf_{h \in \mathbb{H}^\infty} \sup_\theta \left(g - \frac{|a|^2}{w} + w\left|\frac{a}{w} + h\right|^2\right) \quad (2.31)$$

and define a non-negative index $\lambda$ :

$$\lambda = \inf_{h \in \mathbb{H}^\infty} \sup_\theta w\left|\frac{a}{w} + h\right|^2 / \left(\gamma^* - g + \frac{|a|^2}{w}\right) \quad (2.32)$$

it is known that if the function $\Gamma(z, f)$ is of circular form, i.e. $w = \frac{|a|^2}{g}$, the index $\lambda = 1$. If $\Gamma(z, f)$ is not a circular functions, Equation 2.32 can be re-arranged to be

$$\lambda = \inf_{h\in\mathbb{H}^\infty} \sup_{\theta} \left( \frac{w}{\gamma^* - g + \frac{|a|^2}{w}} \right) \left| \frac{a}{w} + h \right|^2 \tag{2.33}$$

$$= \inf_{h\in\mathbb{H}^\infty} \sup_{\theta} w' \left| \frac{a}{w} + h \right|^2 \tag{2.34}$$

where $w' = w/\left(\gamma^* - g + \frac{|a|^2}{w}\right)$.

It is observed from Equation 2.24 that if $w = A$ is selected, we have

$$\lambda = \inf_{h\in\mathbb{H}^\infty} \sup_{\theta} \left( \frac{A}{\gamma^* - g + \frac{|a|^2}{A}} \right) \left| \frac{a}{A} + h \right|^2 \tag{2.35}$$

In the previous discussion, it is known that the problem can be solved directly by the Nehari-commutant lifting formula if $\lambda = 1$, which indicates that the function $\Gamma$ is in the circular form. If $\lambda^* \neq 1$, we know that the function $\Gamma$ is not circular. Thus there is no solution to $OPTd$ problem by the Disk Iteration algorithm. The idea to deal with these functions is proposed in [46] by decreasing $\gamma$ if $\lambda^* < 1$ and increasing $\gamma$ if $\lambda^* > 1$ in terms of choosing different values of $\gamma$ in the line search scheme. The details of finding the appropriate $\gamma$ are described by means of the program codes in the following.

### 2.2.4 Implementation of the Disk Iteration Method

The computer program to implement the DI method in the ANOPT package in Mathematica [41] is translated to the equivalent Matlab programs in this thesis. The structure of the Matlab programs is illustrated in Figure 2.4. Details of the implementation are described in the following :

Suppose we are given a function $\Gamma(z, f(z))$ continuous on $\partial D$ and its derivatives are denoted as $g = \Gamma(z, f(z))$, $a = \frac{\partial}{\partial z}\Gamma(z, f(z))$, and let $H_{h,h}$ be the approximation of the second order terms in its Taylor series, the DI algorithm follows the steps :

1. *Find the descent directional function h*

   (a) **Examine whether the problem is of circular form or not** : an index $square \triangleq g - \frac{|a|^2}{A}$ is used to define the property of the function $\Gamma(z, f(z))$. If $square = 0$, the function $\Gamma(z, f(z))$ is viewed as a circular function and the *Nehari-commutant-lifting formua* is used to find the descent directional function $h$. If $square \neq 0$, it is known that the function $\Gamma(z, f)$ is not of the circular form and an interpolation

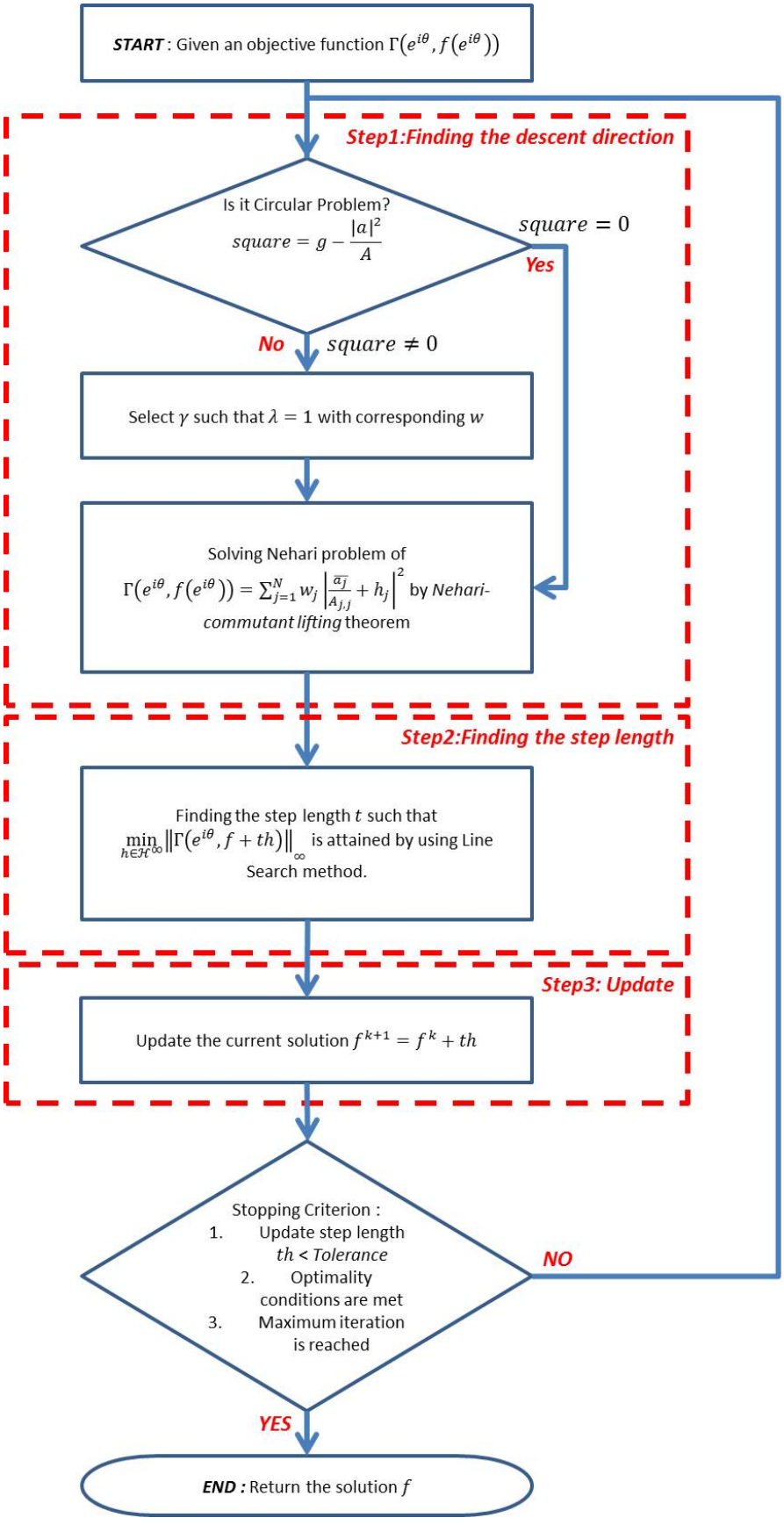



Figure 2.4: Program structure of the DI Method

method is adopted to find an approximate function $w$ so as to obtain the closest circular form of $\Gamma(z, f)$.

(b) **Select $\gamma^*$ such that $\lambda^* = 1$ with corresponding function $w$** : For those functions that result in *square* $\neq 0$, in this step, a function $w$ that approximates $\Gamma(z, f)$ to the nearest circular form is calculated. The procedure to choose the function $w$ such that $\lambda$ is close enough to 1 is illustrated in Figure 2.5 and follows the steps below :

i. *Initialization* : The minimal feasible value of $\gamma$ is calculated to be $\gamma_{min} = g - \frac{|a|^2}{A}$ such that a non-negative index :

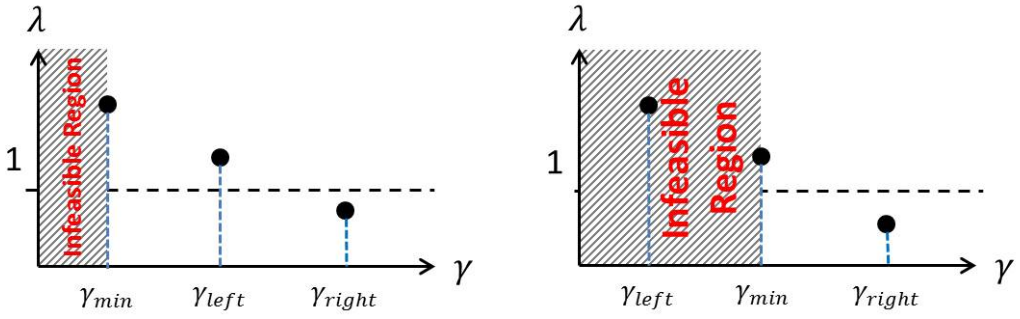
$$\lambda_{min} = \inf_{h \in \mathbb{H}^\infty} \sup_{\theta} \left( \frac{A}{\gamma_{min} - g + \frac{|a|^2}{A}} \right) \left| \frac{a}{A} + h \right|^2$$

and the two extreme values of $\gamma$ by the maximum and minimum of $\Gamma(z, f(z))$ as

$$\gamma_{right} = \max(g) - \frac{|a|^2}{A}$$

$$\gamma_{left} = \min(g) - \frac{|a|^2}{A}$$

we arrive at two possible situations as shown in Figure 2.6, which will be discussed in the next step.




(a) Situation 1 (b) Situation 2

Figure 2.6: Two possible situations

ii. *Search feasible region* : In Figure 2.6, when $\gamma_{left}$ lies on the right hand side of $\gamma_{min}$( see Figure 2.6a), it is known that the possible $\gamma$ such that $\lambda = 1$ appears between $\gamma_{left}$ and $\gamma_{right}$. However, when $\gamma_{left}$ is smaller than $\gamma_{min}$ ( see Figure 2.6b), $\gamma_{left}$ results in the corresponding $\lambda_{left}$ being negative, which is not admissible for the

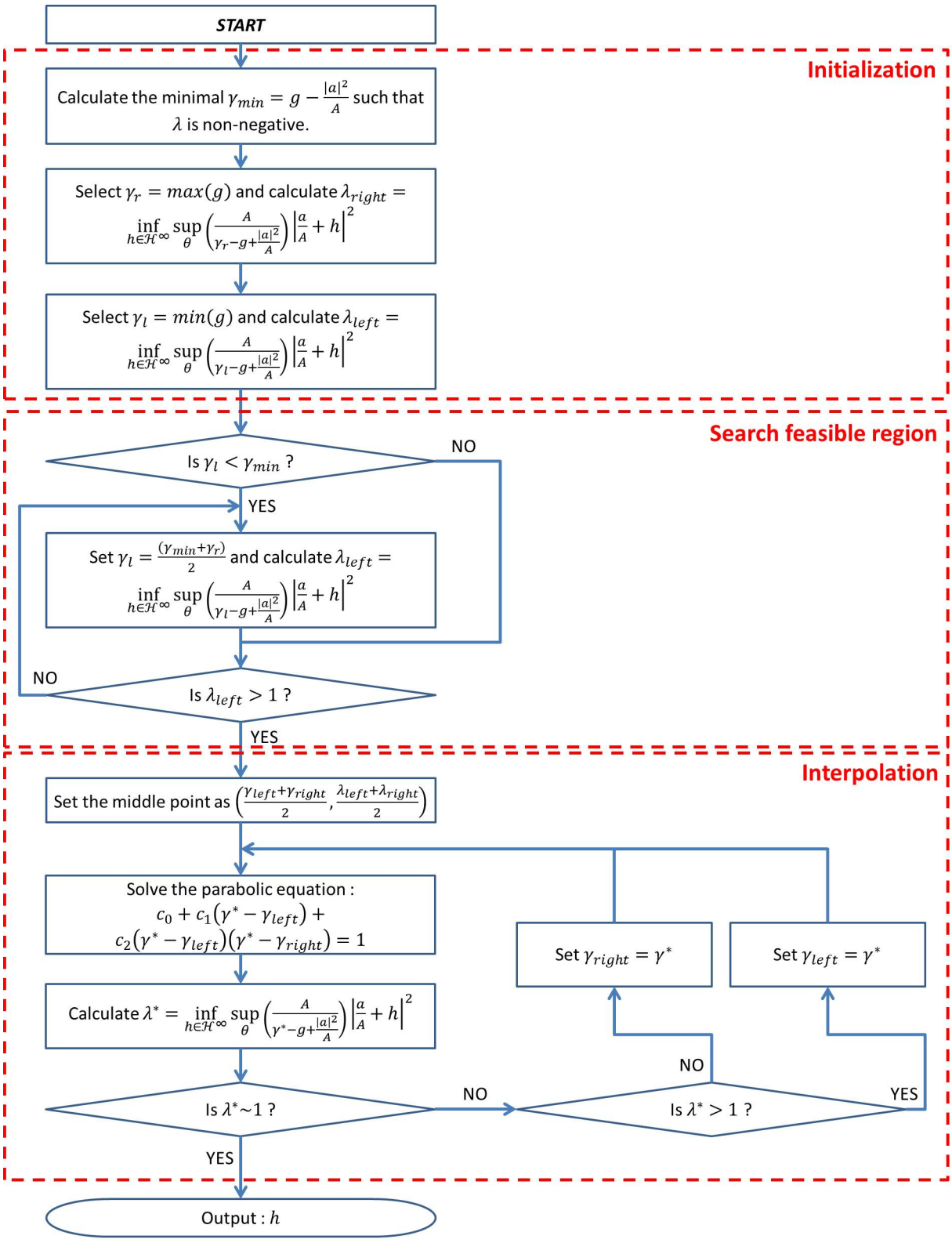



Figure 2.5: Procedure to choose $\gamma^*$ such that $\lambda^* = 1$

problem (of choosing $\gamma^*$ so that $\lambda = 1$). Therefore, in such circumstances, we shift $\gamma_{left}$ to the midpoint of $\gamma_{min}$ and $\gamma_{right}$ , i.e. $\gamma_{left} = \frac{\gamma_{min}+\gamma_{right}}{2}$ , and then compute the corresponding $\lambda_{left}$ . After that, re-setting $\gamma'_{right} = \gamma_{left}$ and $\gamma'_{left} = \frac{\gamma_{min}+\gamma_{left}}{2}$ is performed if this updated $\lambda_{left} < 1$ as shown in Figure 2.7. It is seen that the above procedure ensures $\gamma^*$( so that $\lambda = 1$ ) appears somewhere between $\gamma_{left}$ and $\gamma_{right}$. The region defined in this step will be the starting points of the next step. It is not difficult to observe that the above idea is in fact the same as the well-known *line search* method.

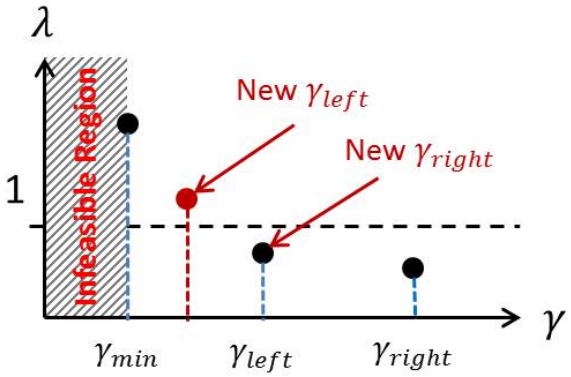



Figure 2.7: Procedure to search the feasible interval $\gamma_{left}$ and $\gamma_{right}$

iii. *Interpolation* : At this point, from the previous results, we obtain the range of the possible $\gamma$s that may result in $\lambda = 1$. In this step, we choose a second-order polynomial function to interpolate the interior points between $\gamma_{left}$ and $\gamma_{right}$ and then approach the analytic optimal point $(\gamma^*, \lambda = 1)$ by reducing the interval from one side. The procedure is illustrated in Figure 2.8.
In Figure 2.8, the green dotted line represents the real distribution of the function with the optimum at the green point of$(\gamma^*, \lambda^*)$and the red dotted lines are the interpolation functions in terms of the second-order polynomials. Given the two points $(\gamma_{left}, \lambda_{left})$ and $(\gamma_{right}, \lambda_{right})$, the quadratic spline function is uniquely determined by using the three points $(\gamma_{left}, \lambda_{left})$ , $(\gamma_{right}, \lambda_{right})$ and $(\gamma^*, \lambda^*)$, which is computed as $\left(\frac{\gamma_{left}+\gamma_{right}}{2}, \frac{\lambda_{left}+\lambda_{right}}{2}\right)$ in the first iteration or the solution point $x$ to the equation below in other iterations :

$$
\begin{aligned}
f(x) &= \lambda_{left} + \frac{(\lambda_{right} - \lambda_{left})}{(\gamma_{right} - \gamma_{left})}(x - \lambda_{left}) \\
&\quad + \frac{(\lambda^* - \lambda_{left})(\gamma_{right} - \gamma_{left}) - (\lambda_{right} - \lambda_{left})(\gamma^* - \gamma_{left})}{(\gamma^* - \gamma_{left})(\gamma^* - \gamma_{right})(\gamma_{right} - \gamma_{left})} \cdot \\
&\quad (x - \gamma_{left})(x - \gamma_{right}) \qquad (2.36) \\
&= 1 \qquad (2.37)
\end{aligned}
$$

( see Appendix C for the derivation ) where

$$
\lambda^* = \inf_{h \in \mathbb{H}^\infty} \sup_{\theta} \left( \frac{A}{\hat{x} - g + \frac{|a|^2}{A}} \right) \left| \frac{a}{A} + h \right|^2
$$

and $\hat{x}$ is the solution to Equation 2.37 in the previous iteration. If the value of $\lambda^*$ is found to be greater than 1 ( as shown at the left of Figure 2.8 ), the new range is narrowed by shifting $(\gamma_{left}, \lambda_{left})$ to the new point $(\gamma^*, \lambda^*)$. On the other hand, if $\lambda^* < 1$ ( as shown at the right of Figure 2.8 ), the right point moves to the new point $(\gamma^*, \lambda^*)$. The iteration then stops when $\lambda$ is close enough to 1, i.e. $|\lambda^* - 1| < \varepsilon$ where $\varepsilon$ is the pre-determined tolerance.
The descent directional function $h$ is hereafter obtained at the same time as finding $\gamma$.

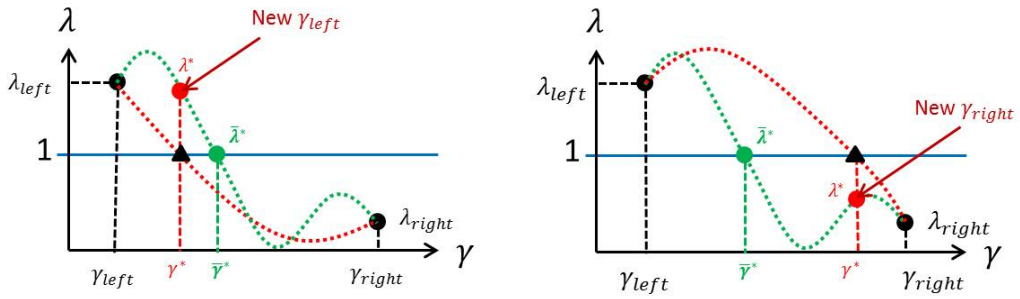



Figure 2.8: Interpolation of $\gamma^*$

(c) **Solving the Nehari problem** :
To apply the Nehari-commutant lifting fomula in order to solve the $OPTc$ problem by the DI method, there are two steps which are implemented in the Matlab programs to compute the best approximation to $\mathbb{H}^\infty$.

i. *Spectral factorization* : As discussed in Section 2.2.1, because the solution to the Nehari problem only exists for the $OPTc$ problem,

the objective function $\Gamma(z, f)$ in the quasi-circular form requires the spectral factorization on $w(z)$ so that

$$\begin{aligned} \Gamma(z, f(z)) &= w(z)\,|k(z) - f(z)|^2 \\ &= \left|\frac{k(z)}{\sigma(z)} - \frac{f(z)}{\sigma(z)}\right|^2 \end{aligned} \tag{2.38}$$

where $\sigma(z)$ is the spectral factor of $w(z)$ such that
$w(z) = \sigma(z) \cdot \sigma^*(z)$.
As the results, the function can be re-arranged in the circular form

$$\Gamma(z, f(z)) = \left|\tilde{k}(z) - \tilde{f}(z)\right|^2 \tag{2.39}$$

where $\tilde{k}(z) = \frac{k(z)}{\sigma(z)}$ and $\tilde{f} = \frac{f(z)}{\sigma(z)}$.
The discussion in Section 2.2.1 gives more details about the NP spectral factorization method.

ii. *Solution to the Nehari Problem* : It is known that the *OPTc* problem can be solved by the Nehari-commutant lifting formula in Section 2.2.2. The practical procedure for solving the Nehari problem implemented in the computer code is derived in [39] as the Nehari commutant-lifting formula :

$$P_{\mathbb{H}^2}\left[\bar{k} P_{\mathbb{H}^{2\perp}}[k\alpha]\right] = \tau^2 \alpha \tag{2.40}$$

where $\alpha$ is a function such that the Hankel operator $\mathcal{H}_k : \mathbb{H}_1^2 \to \mathbb{H}_N^{2\perp}$ with action $\alpha \to P_{\mathbb{H}_N^{2\perp}}(k\alpha)$ attains the norm

$$\|\mathcal{H}_k\| = \operatorname{dist}(k, \mathbb{H}_N^\infty)$$

and

$$\tau = \operatorname{dist}(k, \mathbb{H}_N^\infty)$$

The solution to Equation 2.40 can be approached by updating of the solution $\tilde{f}$ [47] by

$$\alpha_{k+1} = \frac{\alpha_+}{\|\alpha_+\|_\infty}$$

where

$$\alpha_+ = P_{\mathbb{H}^2}\left[\bar{k} P_{\mathbb{H}^{2\perp}}[k\alpha]\right]$$

and $P_{\mathbb{H}^2}[\alpha]$ is the projection operation onto $\mathbb{H}^2$ space, which elim-

inates the negative Fourier coefficients of the function $\alpha$.

The above iteration stops when the optimal $\alpha^*$ is reached as determined by examining if the update increment is smaller than the desired tolerance, i.e. $\|\alpha_{k+1} - \alpha_k\|_\infty < \varepsilon$.

As a result, the best $\mathbb{H}^\infty$ approximation in Equation 2.13 is computed by

$$\tilde{f} = k - \frac{P_{\mathbb{H}^{2\perp}}[\alpha]}{\alpha}$$

and

$$f(z) = \sigma(z) \cdot \tilde{f}(z)$$

2. *Finding the step length t*

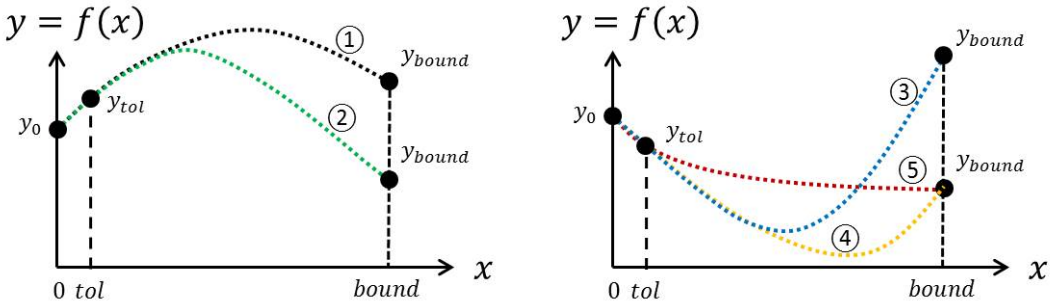
Once the descent directional function $h$ is computed, the next task in the Disk Iteration method is to determine the step length $t \in \mathbb{R}^+$. The step length $t$ can be found by solving the optimization problem :

$$\inf_{f,h \in \mathbb{H}^\infty} \sup_\theta \left\| \Gamma\left(e^{j\theta}, f + th\right) \right\|_{\mathbb{H}^\infty}$$

by the *Golden Section Search* (GSS) method or by the *Sequential Quadratic Programming* (SQP) method. They are discussed respectively in the following.

(a) *Golden Section Search* : The GSS method is coded in the original Mathematica ANOPT program for searching the maximum ( or minimum ) of a concave ( or convex ) function in a determined interval. The idea of the GSS method is in principle the same as the line search method by reducing the feasible interval. The GSS method is described in detail in Appendix D. In the Mathematica ANOPT and Matlab version programs, five different cases are considered in the pre-determined search interval (default $=[0, bound]$), as shown in Figure 2.9. It is found that not all of them need to adopt the GSS method for calculating the step length. These five cases are discussed in the following :

Suppose the feasible search interval is initially defined as $[0, bound]$ and assume there is at most one minimal point in this interval, i.e. the function is convex or concave,




(a) Illustration of two situations

(b) Illustration of three situations

Figure 2.9: Illustration of the different situations in the Golden Section Search method

i. *Case 1 and Case 2* :
By comparing the function value $f(x)$ at the lower bound $x = 0$ and the point $x = tol$ where $tol$ is a small enough distance ( default $= 0.01 * bound$ ) to the lower bound, if $f(x = tol) > f(x = 0)$, we define the first case if the function $f(x = bound)$ at the upper bound is larger than the value $f(x = 0)$ at the lower bound. On the other hand, it is defined as the second case if the function $f(x = bound)$ is smaller than $f(x = 0)$. It is observed that the function $f(x)$ in both cases are increasing at the point of the lower bound. This implies the consequence that the minimal point must be at $x = 0$ in *Case 1* and $x = bound$ in *Case 2*. Therefore, in *Case 1* and *Case 2*, the GSS method will not be activated for calculating the step length. *Case 1* and *Case 2* are shown in Figure 2.9a.

ii. *Case 3 and Case 4* :
If the function around the lower bound ($x = 0$) is decreasing, i.e. $f(x = tol) < f(x = 0)$, it is understood that there may be a minimal point in the interval. In other words, either the function value $f(x = bound)$ at the upper boud is greater ( defined as *Case 3* ) or smaller ( defined as *Case 4* ) than the value $f(x = 0)$ at the lower bound. In the two cases, the GSS method will be adopted to approach the minimum point. The exemption to *Case 4* is when its gradient is continuously decreasing to the upper bound, which is defined as *Case 5* in Figure 2.9. The three cases are illustrated in Figure 2.9b. In *Case 3* and *Case 4*, the GSS method is used to find the local minimum in the interval $[0, bound]$. The algorithm for the GSS method in the Matlab program is described in depth

in Appendix D.

iii. *Case 5* :
The situation when the gradient at the lower bound is negative but there is no minimal point in the interval as shown in Figure 2.9b. It is observed that the minimal point is the one at the upper bound and, in this case, there is no need to use the GSS method.

(b) *Minimax problem* : An alternative solution to the minimization problem of $\inf_{f,h\in\mathbb{H}^\infty} \sup_\theta \left\|\Gamma\left(e^{j\theta}, f+th\right)\right\|_{\mathbb{H}^\infty}$ in Step2 is available by using the standard codes in Matlab function - `fminimax`. Because the minimization problem is discretized at each sampling point, the problem can be viewed as a standard minimax problem, i.e.

$$\min_{t\geq 0} \max_\theta \left\|\Gamma\left(e^{j\theta}, f+th\right)\right\|$$

Compared to the GSS method, the `fminimax` algorithm solves the problem without constraining the search limit. However, it may require more computing effort due to the complexity of *Sequential Quadratic Programming* $(SQP)$ [64] in the `fminimax` function.

3. *Update the solution by* $f^{k+1} = f^k + th$
This is the final step in the DI method where the solution is updated in terms of $f^{k+1} = f^k + th$. If the optimality conditions for the $OPTe$ problem are met for the current solution $f^k$ or the maximum iteration is reached, the DI algorithm is terminated and the code exports the final solution as $f^k$. Otherwise, the updated function $f^{k+1}$ is entered as an input function to Step1 to start the next iteration.

## 2.3 Conclusions

- The optimization problem for the disk iteration method, i.e. the $OPTd$ problem, in [46, 47] is described.
- For numerical and analytic purposes, the transformation from the frequency domain onto the unit circle may be obtained by a linear fractional transform. the $OPT$ problem in the continuous frequency domain is transformed to the $OPTe$ problem on the unit circle. It can be observed in this chapter that many advantages of this transformation to the unit circle become obvious.

- The properties of the sublevel set $S_\theta$ are closely related to the properties of solutions to the $OPTe$ problem. The $OPTc$ problem is then formulated.

- In order to apply the Nehari-commutant lifting formula to the solution to the $OPTd$ problem as a NP method , NP spectral factorization methods are introduced. Wilson's method is used as the main spectral factorization method in the program for its smaller computing effort. The Harris-Davis method is also coded as an alternative spectral factorization method for use in the case when the main method fails to converge.

- The Nehari commutant lifting theorem is applied for solving the Nehari problem, which is proved to be equivalent to the $OPTc$ problem. If the objective function is of circular form in the $OPTe$ problem, it immediately relates to the standard $OPTc$ problem, which is solvable by using the Nehari commutant lifting formula. However, if it is not of circular form, the approximation to the closest circular form is needed. The objective function $\Gamma(z, f)$ is approximated to the circular form in terms of finding the corresponding function $w$ in the $OPTd$ problem such that the index $\lambda = 1$, as shown in Equation 2.32. Searching such $w$ is done by the interpolation method that narrows the lower and upper bounds of the search region.

- The algorithm of the disk iteration method is described in details in three steps. The first step is to find the largest descent directional function $h$. The solution to the Nehari problem is used but not applicable to the optimization problem if the objective function is not in the circular form. Therefore, finding the closest circular form of the objective function is essential. The approximation of the objective function to the circular form is implemented in terms of the quadratic spline interpolation method. The second step is to determine the step length $t$ by using the golden section line search method or the standard code `fminimax` in Matlab. The final step is to update the current solution function $f^k$. The repetition of Step1 and Step2 continues if, in the final step, the stopping criterion is not satisfied.

# Chapter 3

# Newton Iteration Method for $\mathbb{H}^\infty$ Optimization

In the previous chapter, the $OPT$ problem in the frequency domain is transformed to the $OPTe$ problem on the unit circle. The $OPTe$ problem is further related to the $OPTc$ problem due to the relating properties of the sublevel sets $S_\theta(c) = \{\Gamma(z, f(z)) \leq c, c \in \mathbb{R}^+\}$ and the solutions $f$ as discussed in Section 2.1. In this chapter, another method in the spirit of Newton iteration to solve the $OPTc$ problem is presented by using some techniques in functional operator theory. The method is named as Newton Iteration (NI) method in the thesis.

The theoretical descriptions in this chapter are mostly originated from Helton's work in [39]. The chapter describes the implementation of the Helton-Merino-Walker Newton iteration (NI) method for sup-norm optimisation over analytic functions as described in [39]. In Section 3.1, the optimality conditions to the $\mathbb{H}^\infty$ optimization problem is presented. The derivation of the operator equation for the sup-norm optimisation problem is also presented in Secion 3.2.1. The algorithm for the NI method proposed in [41, 47] is also outlined in Section 3.2.1. In Section 3.2.2, the matrix representation of the Jacobian operator in the NI algorithm is described as required for the Matlab implementation. The program code to implement the NI method is thus written in Matlab.

## 3.1 Optimality Conditions to the Optimization Problem

In this section, the optimality conditions to the $OPTc$ problem are presented. The typical example of the $OPTc$ problem is the Nehari problem [70] where the objective function $\Gamma(z, f^*(z))$ is given as $\Gamma(z, f^*(z)) = |k(z) - f^*(z)|^2$ with

$\gamma^* = \inf_{f^* \in \mathbb{A}} \sup_{\omega} \Gamma(z, f^*(z))$ and $f^*(z)$ is viewed as the *best* $\mathbb{H}^\infty$ *approximation* to the function $k(z)$. The optimality test for the Nehari problem was proposed by Adamajan et al [3] and extened by Poreda [74] to be

**Theorem 5.** *$f^*(z)$ is the optimal solution to the Nehari problem if and only if*

- *if $f^*(z) \in \mathbb{A}$, $|k(z) - f^*(z)|$ is a constant for all $z = e^{j\theta}$*
- *the winding number of $(k^*(z) - f^*(z))$ about 0 is negative*

Furthermore, it is shown in [45] that the sublevel sets of the objective function $\Gamma(z, f(z))$: $S_\theta(c) = \{\Gamma(z, f(z)) \leq c, c \in \mathbb{R}^+\}$ have the shape of disk and their properties of boundness, simple connectness, smoothness and convexity are closely related to the properties of solutions to the $OPTe$ problem. This addressed the problem of dimension $N = 1$, i.e. $f(z) \in \mathbb{C}$. For higher dimensions, Helton [43] generalized Poreda's optimality conditions to the theorem for $N > 1$ [43] :

**Theorem 6.** *Given the generic $\Gamma(z, f(z))$ and a continuous function $f(z) \in \mathbb{A}$ , if the gradient $\frac{\partial \Gamma}{\partial z}(z, f(z))$ is also continous and never vanishes [see Appendix A], the solution $f^*(z)$ is the strictly local directional optimizer to the $OPTc$ problem if and only if*

- *$\Gamma(z, f^*(z))$ is constant for all $z = e^{j\theta}$*
- *the winding number of $\frac{\partial \Gamma}{\partial z}(z, f^*(z))$ about 0 is positive*

*where the strictly local directional optimizer $f^*(z)$, which is also the local optimum, is defined for $h(z) \in \mathbb{A}$ and $\alpha \in \mathbb{R}^+$ as*

$$\sup_{\theta} \Gamma(z, f^*(z)) < \sup_{\theta} \Gamma(z, f^*(z) + \alpha h(z)) \tag{3.1}$$

Merino [68] then further proved that the winding number of $\frac{\partial \Gamma}{\partial z}(z, f^*(z))$ in the theorem is one. More about the uniqueness, continuity and existence of the solution to the $OPTc$ problem for $N = 1$ can be found in [45]. However, the above conditions do not hold for the matrix-valued cases ( i.e. $N > 1$ ) [38].

To extend the optimality conditions to more general case ( $N \geq 1$ ), the generalization of the Corona theorem proposed in [43] is required. From the observation in the dual extremal method [28], the extension of the second condition given above was proved in [43] and further extended to general cases by Helton et al. [48]. It is concluded in [48] that the neccessary and sufficient conditions for the $OPTe$ problem are

**Theorem 7.** *Given a function* $f(z) \in \Omega$ *where* $\Omega$ *denotes the set* $\Omega = \left\{ f \in \mathbb{A}_N : \int \langle f, G \rangle \frac{d\theta}{2\pi} = \int_0^{2\pi} f^T \bar{G} \frac{d\theta}{2\pi} = c \right\}$ *and* $c \in \mathbb{C}$ *is a constant and a set* $\Omega_0 = \left\{ f \in \mathbb{A}_N : \int_0^{2\pi} f^T \bar{G} \frac{d\theta}{2\pi} = 0 \right\}$, *assume that* $\frac{\partial \Gamma}{\partial z}(z, f(z))$ *does not vanish on* $\partial D$ *and a continuous positive-valued function* $\Gamma(z, f(z))$ *is at least three times differentiable on* $\partial D \times \mathbb{C}^N$, *the function* $f^*$ *is a directional solution to the* $OPTe$ *problem for which*

- $\Gamma(z, f^*) = \gamma^*$ *where* $\gamma^* \in \mathbb{R}$ *is a positive constant*
- *there exists a positive-valued measurable function* $\lambda^* : \partial \to \mathbb{R}^+$, *functions* $F^*, G \in \mathbb{RH}_N^1$ *and constants* $\kappa^* \in \mathbb{R}$ *such that*

$$\frac{\partial \Gamma}{\partial z}(z, f^*) = \lambda^* \left( e^{j\theta} F^* + \sum \kappa^* G \right)$$

- *If a subspace* $\mathbb{N} = \left\{ h \in \Omega_0 : \Re\left( \frac{\partial}{\partial z} \Gamma(\cdot, f^*) \right) = 0 \right\}$,

$$\sup_\theta \left\{ \bar{h^T} \frac{\partial^2}{\partial z \partial \bar{z}} \Gamma(\cdot, f^*)\, h + \Re\left( h^T \frac{\partial^2}{\partial z^2} \Gamma(\cdot, f^*)\, h \right) \right\} \geq 0, \forall h \in \mathbb{N} \diagdown \{0\}$$

To study the solution to the $OPTc$ problem, we first consider the related problem in terms of the $\mathcal{L}_1$ norm [39] :

$OPT_1$ *Given a continuous positive valued function* $\Gamma$ *in* $D$, *and a set* $\mathbb{A}$ *of the analytic functions on the unit circle, find the optimal function* $f^*(z)$ *such that*

$$\gamma = \inf_{f^* \in \mathbb{A}} \sup_\theta \left\| \Gamma(z, f^*(z)) \right\|_{\mathcal{L}_1} = \inf_{f^* \in \mathbb{A}} \sup_\theta \int_0^{2\pi} \Gamma(z, f^*(z)) \cdot \frac{d\theta}{2\pi} \quad (3.2)$$

*where* $z$ *are the points on the unit circle, and* $f^*(z)$ *is a continuous function in* $\mathbb{H}^\infty$.

The optimality conditions to the $OPT_1$ problem proved in [39] are expressed by the following theorem :

**Theorem 8.** *Let* $\Gamma(z, f) \in \mathbb{C}^N$ *and* $f^* \in \mathbb{H}_N^\infty$ *be a local directional optimizer to* $OPT_1$ *problem such that* $\frac{\partial}{\partial z} \Gamma(z, f^*(z))$ *never vanishes on* $\partial D$.

*The following conditions hold :*

*I.* *There exists a function* $F \in \mathbb{H}_N^1$ *such that*

$$\frac{\partial}{\partial z} \Gamma(z, f^*) = e^{j\theta} F \quad (3.3)$$

*II.* *For every nonzero* $h \in \mathbb{H}^{\infty}$,

$$\sup_{\theta} \left\{ \bar{h}^t \frac{\partial^2}{\partial z \partial \bar{z}} \Gamma\left(z, f^*\right) h + \Re\left( h^T \frac{\partial^2}{\partial z^2} \Gamma\left(z, f^*\right) h \right) \right\} \geq 0 \qquad (3.4)$$

It is discussed in [39] that the second condition *II* is always true if the function $\Gamma\left(z, f\left(z\right)\right)$ is strongly convex. Therefore, for some cases, the conditions *I* is sufficient to give the optimizer $f^*$.

Moreover, the optimality conditions to the $OPT_{\infty}$ problem in [38] are

$OPT_{\infty}$ *Let* $\Gamma\left(z, f\left(z\right)\right) \in \mathbb{C}^N$ *and* $f^* \in \mathbb{H}_N^{\infty}$ *be a local directional optimizer to the OPTe problem such that* $\frac{\partial}{\partial z}\Gamma\left(z, f^*\left(z\right)\right)$ *never vanishes on* $\partial D$. *The following conditions hold :*

*I.* $\Gamma\left(z, f^*\right)$ *is constant on* $\partial D$

*II. There exists a function* $F \in \mathbb{H}_N^1$ *and a positive and measurable linear functional* $\lambda$*:* $\partial \rightarrow \mathbb{R}^+$ *on* $\partial D$ *such that*

$$\lambda^{-1} \frac{\partial}{\partial z} \Gamma\left(z, f^*\right) = e^{j\theta} F$$

*III. For every nonzero* $h \in \mathbb{H}^{\infty}$,

$$\sup_{\theta} \left\{ \bar{h}^t \frac{\partial^2}{\partial z \partial \bar{z}} \Gamma\left(z, f^*\right) h + \Re\left( h^T \frac{\partial^2}{\partial z^2} \Gamma\left(z, f^*\right) h \right) \right\} \geq 0$$

*where* $z$ *are the points spaced around the unit circle, and* $f^*\left(z\right)$ *is a continuous function in* $\mathbb{H}^{\infty}$

In observing the optimality conditions for the $OPT_1$ problem and the $OPT_{\infty}$ problem, we see the similarity between the first condition for the $OPT_1$ problem and the second condition for the $OPT_{\infty}$ problem. The connection between the solution to the $OPT_1$ problem and the $OPT_{\infty}$ problem is obvious and proved in [39]. In some cases, it allows the algorithm to solve the $OPT_{\infty}$ problem by solving $OPT_1$ problem. Furthermore, if the function $\Gamma\left(z, f^*\right)$ is strongly convex, the computer program can be accelerated [46] by simply meeting the first condition for the $OPT_1$ problem : $\frac{\partial}{\partial z}\Gamma\left(z, f^*\right) = e^{j\theta} F$. However, in the following derivation, we consider the optimality conditions for the $OPT_{\infty}$ problem for purpose of generalization.

In summary, it is mentioned in [39] that the optimality conditions to the $OPT_{\infty}$ problem are formulated as follows :

*Find the tuple* $(f^*, \lambda^*, F^*, \gamma^*)$ *where* $f^* \in \mathbb{H}_N^\infty$, $\gamma^* \in R$, $F^* \in \mathbb{H}_N^1$ *and* $\lambda^* : \partial \to \mathbb{R}^+$ *measurable and nonzeros such that the two conditions*

$$\Gamma\left(z, f^*\right) = \gamma^* \tag{3.5}$$

$$\lambda^{-1} \frac{\partial}{\partial z}\left(z, f^*\right) = \chi F^* \tag{3.6}$$

*are satisfied.*

## 3.2 Newton Iteration Method

### 3.2.1 Derivation and Solution of the Operator Equation

This section interprets the results in [39] for computer implementation by using the operators. Fundamental details of the operators are described in [109]. The operators used in the thesis are defined in terms of the function spaces on the unit circle and unit disk. The following outlines the properties of these operators.

Let $\mathcal{L}_2\left(\partial D\right)$ be the set of the matrix-valued functions $F$ in Hilbert space on the unit circle $\partial D$ :

$$\mathcal{L}_2\left(\partial D\right) = \left\{ F := \frac{1}{2\pi} \int_0^{2\pi} Trace\left(F^*\left(e^{j\theta}\right) F\left(e^{j\theta}\right)\right) d\theta < \infty \right\}$$

and $\mathbb{H}^2$ be the subspace of $\mathcal{L}_2\left(\partial D\right)$ with functions $\tilde{F}$ analytic in $D$ :

$$\mathbb{H}^2\left(\partial D\right) = \left\{ \tilde{F} \in \mathcal{L}_2\left(\partial D\right) := \frac{1}{2\pi} \int_0^{2\pi} \tilde{F}\left(e^{j\theta}\right) e^{j\theta} d\theta = 0, \forall n > 0 \right\}$$

We denote $\mathbb{H}^{2\perp}$ as the orthogonal space of $\mathbb{H}^2$. In other words, invoking the Fourier transform, $\mathbb{H}^2$ space are the functions where their Fourier coefficients whose idexes are greater than zero. On the other hand, all the positive Fourier coefficients of the functions in $\mathbb{H}^{2\perp}$ will vanish. For example, let $F \in \mathcal{L}_2\left(\partial D\right)$ and the Fourier expansion of $F$ be given as

$$F\left(e^{j\theta}\right) \sim \sum_{n=-\infty}^{\infty} F_n \cdot e^{jn\theta}$$

then $F \in \mathbb{H}^2$ if and only if $F_n = 0$, $\forall n < 0$ and $F \in \mathbb{H}^{2\perp}$ if and only if $F_n = 0$, $\forall n \geq 0$ .

By taking the orthogonal projection operator on both sides of Equation 3.5 and 3.6 respectively, since $\gamma^*$ is constant and $F^* \in \mathbb{H}_N^1$, the projections $P_{\mathbb{H}_1^{2\perp}}[\gamma^*] = 0$ and $P_{\mathbb{H}_N^{2\perp}}[F^*] = 0$. Further let $\mathbb{H}_{N,0}^{2\perp} \triangleq \chi \mathbb{H}_N^{2\perp}$ then the projection of $\chi F^*$ onto the space $\mathbb{H}_{N,0}^{2\perp}$ gives $P_{\mathbb{H}_{N,0}^{2\perp}}[\chi F^*] = 0$. We therefore have

$$P_{\mathbb{H}_1^{2\perp}}[\Gamma(\cdot, f^*)] = 0 \tag{3.7}$$

$$P_{\mathbb{H}_{N,0}^{2\perp}}\left[\lambda^{-1}\frac{\partial}{\partial z}\Gamma(\cdot, f^*)\right] = 0 \tag{3.8}$$

Let $\lambda^{-1}$ be defined as $\lambda^{-1} = 1 + 2\Re(\chi\beta)$ where $\beta$ is a scalar-valued analytic function on $\partial D$, the *Operator Equation* is derived as

$$T\begin{pmatrix} f^* \\ \beta^* \end{pmatrix} \triangleq \begin{pmatrix} P_{\mathbb{H}_{N,0}^{2\perp}}\left[(1 + 2\Re(\chi\beta))\frac{\partial}{\partial z}\Gamma(\cdot, f^*)\right] \\ P_{\mathbb{H}_1^{2\perp}}[\Gamma(\cdot, f^*)] \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.9}$$

We now consider the solution of the operator equation. It is known that any analytic function $f(x)$ can be expanded by the Taylor series about $x_n$ as

$$f(x_n + \Delta x) = f(x_n) + f'(x_n)\Delta x + \frac{1}{2}f''(x_n)\Delta x^2 + \cdots \tag{3.10}$$

The function $f(x_n + \Delta x)$ approaches the critical point when the first derivative is close to zero ( i.e. $f'(x^*) = 0$ ). Hence, by differentiating Equation 3.10, we get $f'(x_n) + f''(x_n)\triangle x = 0$. As a result, the Newton-Iteration formula is derived as

$$x_{n+1} = x_n + \Delta x = x_n - \frac{f'(x_n)}{f''(x_n)} \tag{3.11}$$

If the two projected functions in Equation 3.7 and 3.8 are expanded up to second order as

$$\Gamma(\cdot, f + \varepsilon) = g + 2\Re\left(a^t\varepsilon\right) + \bar{\varepsilon}^T A\varepsilon + \Re\left(\varepsilon^T B\varepsilon\right) + \cdots \tag{3.12}$$

$$\frac{\partial}{\partial z}\Gamma(\cdot, f + \varepsilon) = a + A\bar{\varepsilon} + B\varepsilon + \cdots \tag{3.13}$$

where $g = \Gamma(\cdot, f)$,$a = \frac{\partial}{\partial z}\Gamma(\cdot, f)$, $A = \frac{\partial^2}{\partial z \partial \bar{z}}\Gamma(\cdot, f)$, $B = \frac{\partial^2}{\partial z^2}\Gamma(\cdot, f)$,
the operator equation is then reformulated as

$$T\begin{pmatrix} f+\varepsilon \\ \beta+\delta \end{pmatrix} = \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi(\beta+\delta)\right)\right)\frac{\partial}{\partial z}\Gamma\left(\cdot, f+\varepsilon\right)\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[\Gamma\left(\cdot, f+\varepsilon\right)\right] \end{pmatrix} \tag{3.14}$$

By expanding $T\begin{pmatrix} f+\varepsilon \\ \beta+\delta \end{pmatrix}$ in the Taylor series up to the first order, Equation 3.14 becomes

$$T\begin{pmatrix} f \\ \beta \end{pmatrix} + T'_{f,\beta}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} = 0 \tag{3.15}$$

where $T'_{f,\beta}$ denotes the Jacobian operator, which, in principle, is the derivative of $T$ with respect to $(f,\beta)^T$ .
This, therefore, allows the use of Equation 3.11 to update the solution $f$ at the $k$-th iteration by

$$\begin{pmatrix} f^{k+1} \\ \beta^{k+1} \end{pmatrix} = \begin{pmatrix} f^{k} \\ \beta^{k} \end{pmatrix} - T'^{-1}_{f,\beta} T\begin{pmatrix} f^{k} \\ \beta^{k} \end{pmatrix} \tag{3.16}$$

To this point, the next question is how to find out the Jacobian operator $T'_{f,\beta}$. We start the derivation by rewriting Equation 3.15

$$T\begin{pmatrix} f+\varepsilon \\ \beta+\delta \end{pmatrix} = T\begin{pmatrix} f \\ \beta \end{pmatrix} + T'_{f,\beta}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} \tag{3.17}$$

and rearrange to

$$T'_{f,\beta}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} = T\begin{pmatrix} f+\varepsilon \\ \beta+\delta \end{pmatrix} - T\begin{pmatrix} f \\ \beta \end{pmatrix} \tag{3.18}$$

Substituting Equation 3.9 and Equation 3.14 into Equation 3.18 and ignoring all terms that are of high order (e.g. $2\Re\left(\chi\delta\right)\cdot A\bar{\varepsilon}$, $\bar{\varepsilon^t}A\varepsilon$, ... etc), the Jacobian operator is formulated as

$$\begin{aligned} T'_{f,\beta}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix} &= \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi(\beta+\delta)\right)\right)\frac{\partial}{\partial z}\Gamma\left(\cdot, f+\varepsilon\right)\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[\Gamma\left(\cdot, f+\varepsilon\right)\right] \end{pmatrix} && (3.19) \\ &\quad - \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi\beta\right)\right)\frac{\partial}{\partial z}\Gamma\left(\cdot, f\right)\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[\Gamma\left(\cdot, f\right)\right] \end{pmatrix} && (3.20) \\ &= \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi\beta\right)\right)\left(A\bar{\varepsilon}+B\varepsilon\right)+2a\Re\left(\chi\delta\right)\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[2\Re a^T\varepsilon\right] \end{pmatrix} && (3.21) \end{aligned}$$

It is pointed out in [39] that Equation 3.21 can be further rearranged in the compact form ( refer to *Appendix B* for more details )

$$T'_{f,\beta} = \begin{pmatrix} I_N & 0 \\ 0 & \bar{\chi} \end{pmatrix} CT_{M_1} + \begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix} \mathcal{H}_{M_2} \tag{3.22}$$

where
$T_{M_1}$ is the Toeplitz operator with symbol

$$M_1 = \begin{pmatrix} \omega A & \chi \bar{a} \\ \bar{\chi} a^T & 0 \end{pmatrix} \tag{3.23}$$

$H_{M_2}$ is the Hankel operator with symbol

$$M_2 = \begin{pmatrix} \bar{\chi}\omega B & a \\ a^T & 0 \end{pmatrix} \tag{3.24}$$

and $I_N$ is the $N \times N$ identity matrix, and $\omega = 1 + 2\Re(\chi\beta)$.

### 3.2.2 Matrix Computation of the Jacobian $T'_{f,\beta}$

In this section, to allow the Matlab implementation of the algorithm, the matrix representation of the operators in Equation 3.22 , which are used to compute the Jacobian $T'_{f,\beta}$, are described.

Denote $P_+ : \mathcal{L}_2(\partial D) \to \mathbb{H}^2$ and $P_- : \mathcal{L}_2(\partial D) \to \mathbb{H}^{2\perp}$ as the respective orthogonal projections onto $\mathbb{H}^2$ and $\mathbb{H}^{2\perp}$ in $\mathcal{L}_2$. For any function $G_d \in \mathcal{L}_2(\partial D)$, the Toeplitz operator and the Hankel operator applying on the function $G_d$ are defined as

$$T_{G_d} = P_+ G_d$$

and

$$\mathcal{H}_{G_d} = P_- G_d$$

In other words, the Toeplitz operator is the mapping from $\mathcal{L}_2(\partial D)$ to $\mathbb{H}^2$ and the Hankel operator is the mapping from $\mathcal{L}_2(\partial D)$ to $\mathbb{H}^{2\perp}$. In terms of z-transfrom and the frequency response of $G_d(z) = \sum_{n=-\infty}^{\infty} G_n \cdot z^n$ , we then have a z-transform transfer function and $G_d\left(e^{j\theta}\right) = G_d(z)$, $G_d\left(e^{j\theta}\right) = \sum_{n=-\infty}^{\infty} G_n \cdot e^{jn\theta}$ in $\mathcal{L}_2(\partial D)$.

Thus in $\mathcal{L}_2(\partial D)$, for an input $u\left(e^{j\theta}\right) = \sum_{n=-\infty}^{\infty} u_n \cdot e^{jn\theta}$ and an output

$y\left(e^{j\theta}\right) = \sum_{n=-\infty}^{\infty} y_n \cdot e^{jn\theta}$ in $\mathbb{H}^2$, we have

$$y(z) = G_d(z)\, u(z)$$

that is

$$\sum_{n=-\infty}^{\infty} y_n \cdot e^{jn\theta} = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} G_n \cdot u_m \cdot e^{j(n+m)\theta}$$

In infinite matrix form, this is given equivalently by either

$$\begin{bmatrix} \vdots \\ \vdots \\ y_1 \\ y_0 \\ - \\ y_{-1} \\ y_{-2} \\ \vdots \\ \vdots \end{bmatrix} = \left[\begin{array}{cccc|cccc} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\ \cdots & \ddots & \vdots & \vdots & \vdots & \vdots & \iddots & \cdots \\ \cdots & \cdots & G_0 & G_1 & G_2 & G_3 & \cdots & \cdots \\ \cdots & \cdots & G_{-1} & G_0 & G_1 & G_2 & \cdots & \cdots \\ - & - & - & - & - & - & - & - \\ \cdots & \cdots & G_{-2} & G_{-1} & G_0 & G_1 & \cdots & \cdots \\ \cdots & \cdots & G_{-3} & G_{-2} & G_{-1} & G_0 & \cdots & \cdots \\ \cdots & \iddots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right] \begin{bmatrix} \vdots \\ \vdots \\ u_1 \\ u_0 \\ - \\ u_{-1} \\ u_{-2} \\ \vdots \\ \vdots \end{bmatrix}$$

or by

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ - \\ y_{-1} \\ y_{-2} \\ \vdots \\ \vdots \end{bmatrix} = \left[\begin{array}{cccc|cccc} G_0 & G_{-1} & \cdots & \cdots & G_1 & G_2 & \cdots & \cdots \\ G_1 & G_0 & \cdots & \cdots & G_2 & G_3 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ - & - & - & - & - & - & - & - \\ G_{-1} & G_{-2} & \cdots & \cdots & G_0 & G_1 & \cdots & \cdots \\ G_{-2} & G_{-3} & \cdots & \cdots & G_{-1} & G_0 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \end{array}\right] \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ - \\ u_{-1} \\ u_{-2} \\ \vdots \\ \vdots \end{bmatrix}$$

$$= \left[\begin{array}{ccc} T_1 & | & \mathcal{H}_1 \\ - & + & - \\ \mathcal{H}_2 & | & T_2 \end{array}\right] \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ - \\ u_{-1} \\ u_{-2} \\ \vdots \\ \vdots \end{bmatrix}$$

The latter block matrix structure then has the form

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ - \\ y_{-1} \\ y_{-2} \\ \vdots \\ \vdots \end{bmatrix} = \left[\begin{array}{ccc} T_1 & | & \mathcal{H}_1 \\ - & + & - \\ \mathcal{H}_2 & | & T_2 \end{array}\right] \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ - \\ u_{-1} \\ u_{-2} \\ \vdots \\ \vdots \end{bmatrix}$$

The matrices $T_1$ and $T_2$ are known as *(block) Toeplitz matrices* and $\mathcal{H}_1$ and $\mathcal{H}_2$ are called *(block) Hankel matrices*. They are respectively the representation of the *Toeplitz operator* and the *Hankel operator*. These matrix forms are used to represent the *Toeplitz* and *Hankel operators* in the Matlab Newton Iteration method computer program.

In summary, according to Equation 3.22, the Jacobian operator on the update increment function is

$$-T'_{f,\beta}\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix} = T\begin{pmatrix} f\\ \beta\end{pmatrix}$$

which has the form of

$$\left(\begin{pmatrix} I_N & 0\\ 0 & \bar{\chi}\end{pmatrix} CT_{M_1} + \begin{pmatrix}\chi I_N & 0\\ 0 & 1\end{pmatrix} H_{M_2}\right)\left[-\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix}\right] = T\begin{pmatrix} f\\ \beta\end{pmatrix}$$

Denoting the vector $-\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix}$ as $h$, and $T\begin{pmatrix} f\\ \beta\end{pmatrix}$ as $Q$, we can write

$$\left(\begin{pmatrix} I_N & 0\\ 0 & \bar{\chi}\end{pmatrix} CT_{M_1} + \begin{pmatrix}\chi I_N & 0\\ 0 & 1\end{pmatrix} H_{M_2}\right)[h] = Q$$

and by defining component terms $Q_1$ and $Q_2$ of $Q$ by

$$\begin{aligned}\begin{pmatrix} I_N & 0\\ 0 & \bar{\chi}\end{pmatrix} CT_{M_1}[h] &= Q_1\\ \begin{pmatrix}\chi I_N & 0\\ 0 & 1\end{pmatrix} H_{M_2}[h] &= Q_2\end{aligned}$$

we have

$$Q = Q_1 + Q_2$$

In the following sections, we will use such relationships to construct the Jacobian operator and compute its inverse in order to calculate the update increment functions $\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix}$.

In the Matlab computer implementation, firstly the function vector $Q$ is computed from

$$Q = T\begin{pmatrix} f\\ \beta\end{pmatrix} = \begin{pmatrix} P_{H^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi\beta\right)\right)\frac{\partial}{\partial z}\Gamma\left(\cdot,f\right)\right]\\ P_{H^{2\perp}_{1}}\left[\Gamma\left(\cdot,f\right)\right]\end{pmatrix} \tag{3.25}$$

which entails the elimination of all the positive index Fourier coefficients of $\left(1+2\Re\left(\chi\beta\right)\right)\frac{\partial}{\partial z}\Gamma\left(\cdot,f\right)$ and all the non-negative Fourier coefficients of $\Gamma\left(\cdot,f\right)$.

The inverse of $T'_{f,\beta}$ is then found by matrix inversion, and the update function vector $h$ is then obtained from

$$h = T'^{-1}_{f,\beta}[Q] = \left( \begin{pmatrix} I_N & 0 \\ 0 & \bar{\chi} \end{pmatrix} CT_{M_1} + \begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix} H_{M_2} \right)^{-1} [Q] \tag{3.26}$$

#### 3.2.2.1 Matrix Representation of the Conjugate Toeplitz Operator

We may represent the operations with the Toeplitz operator of Equation 3.22 by $y = T_{M_1} h$, which restricted to positive index Fourier coefficients has the matrix form

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} G_0 & G_{-1} & G_{-2} & \cdots & \cdots \\ G_1 & G_0 & G_{-1} & \cdots & \cdots \\ G_2 & G_1 & G_0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ \vdots \end{bmatrix}$$

where

$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \end{bmatrix}$ is a vector that contains the Fourier coefficients of $y$

and $\begin{bmatrix} G_0 & G_{-1} & G_{-2} & \cdots & \cdots \\ G_1 & G_0 & G_{-1} & \cdots & \cdots \\ G_2 & G_1 & G_0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$ is a matrix whose elements are the Fourier coefficients corresponding to the term $T_{M_1}$ of Equation 3.22 whose term is known as the Toeplitz operator with the symbol $M_1$.

In Equation 3.23, we require $M_1 = \begin{pmatrix} \omega A & \chi \bar{a} \\ \bar{\chi} a^T & 0 \end{pmatrix}$ so that $T_{M_1}$ then has the corresponding block matrix representation

$$T_{M_1} = \begin{bmatrix} \begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \begin{bmatrix} G_{-1,11} & G_{-1,12} \\ G_{-1,21} & G_{-1,22} \end{bmatrix} & \begin{bmatrix} G_{-2,11} & G_{-2,12} \\ G_{-2,21} & G_{-2,22} \end{bmatrix} & \cdots & \cdots \\ \begin{bmatrix} G_{1,11} & G_{1,12} \\ G_{1,21} & G_{1,22} \end{bmatrix} & \begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \begin{bmatrix} G_{-1,11} & G_{-1,12} \\ G_{-1,21} & G_{-1,22} \end{bmatrix} & \cdots & \cdots \\ \begin{bmatrix} G_{2,11} & G_{2,12} \\ G_{2,21} & G_{2,22} \end{bmatrix} & \begin{bmatrix} G_{1,11} & G_{1,12} \\ G_{1,21} & G_{1,22} \end{bmatrix} & \begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where

$G_{-1,11}, G_{0,11}, G_{1,11}, \cdots$ are the Fourier coefficients of the function $\omega A$ in $M_1$,
$G_{-1,12}, G_{0,12}, G_{1,12}, \cdots$ are the Fourier coefficients of the function $\chi\bar{a}$ in $M_1$,
$G_{-1,21}, G_{0,21}, G_{1,21}, \cdots$ are the Fourier coefficients of the function $\bar{\chi}a$ in $M_1$,
and $G_{-1,22}, G_{0,22}, G_{1,22}, \cdots$ are the ( all zero ) Fourier coefficients of the function 0 in $M_1$.

The above matrix representation is only for the Toeplitz operator restricted to the positive index Fourier coefficients, and therefore this requires to be extended to include the negative index Fourier coefficients. Now the function mapped by the Toeplitz operator onto $\mathbb{H}^2$ space has only positive non-zero index coefficients since $h_i, \forall i \geq 0$, that is the *Toeplitz matrix* only acts on the positive index Fourier coefficients of the input. Therefore, to represent the full complement of Fourier coefficients, the full Toeplitz matrix equation, $y = T_{M_1} h$, is written thus

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ - \\ y_{-1} \\ y_{-2} \\ \vdots \\ \vdots \end{bmatrix} =$$

$$\left[\begin{array}{ccccc|ccc}
\begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \begin{bmatrix} G_{-1,11} & G_{-1,12} \\ G_{-1,21} & G_{-1,22} \end{bmatrix} & \begin{bmatrix} G_{-2,11} & G_{-2,12} \\ G_{-2,21} & G_{-2,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
\begin{bmatrix} G_{1,11} & G_{1,12} \\ G_{1,21} & G_{1,22} \end{bmatrix} & \begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \begin{bmatrix} G_{-1,11} & G_{-1,12} \\ G_{-1,21} & G_{-1,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
\begin{bmatrix} G_{2,11} & G_{2,12} \\ G_{2,21} & G_{2,22} \end{bmatrix} & \begin{bmatrix} G_{1,11} & G_{1,12} \\ G_{1,21} & G_{1,22} \end{bmatrix} & \begin{bmatrix} G_{0,11} & G_{0,12} \\ G_{0,21} & G_{0,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\
- & - & - & - & - & + & - & - & - \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots
\end{array}\right]$$

$$\cdot \begin{bmatrix} h_0 & h_1 & \cdots & | & h_{-1} & h_{-2} & \cdots \end{bmatrix}^T$$

Next, the conjugate operator $C$ acting on $y$ must be determined. Observing that for a function $F$ represented by the Fourier coefficients $\{\ldots, F_{-2}, F_{-1}, F_0, F_1, F_2, \ldots\}$, the Fourier coefficients of the conjugate of the function $\overline{F}$ is obtained by taking the conjugate of $\{\ldots, F_{-2}, F_{-1}, F_0, F_1, F_2, \ldots\}$ and reversing the order, to give $\left\{\ldots, \overline{F}_2, \overline{F}_1, \overline{F}_0, \overline{F}_{-1}, \overline{F}_{-2}, \ldots\right\}$. Such an operation applied to the Toeplitz matrix in $\tilde{y} = CT_{M_1}h$ results in the form of

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ - \\ y_{-1} \\ y_{-2} \\ \vdots \\ \vdots \end{bmatrix} =$$

$$\left[\begin{array}{ccccc|ccc}
\begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-1,11} & \overline{G}_{-1,12} \\ \overline{G}_{-1,21} & \overline{G}_{-1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-2,11} & \overline{G}_{-2,12} \\ \overline{G}_{-2,21} & \overline{G}_{-2,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
- & - & - & - & - & + & - & - & - \\
\begin{bmatrix} \overline{G}_{1,11} & \overline{G}_{1,12} \\ \overline{G}_{1,21} & \overline{G}_{1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-1,11} & \overline{G}_{-1,12} \\ \overline{G}_{-1,21} & \overline{G}_{-1,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
\begin{bmatrix} \overline{G}_{2,11} & \overline{G}_{2,12} \\ \overline{G}_{2,21} & \overline{G}_{2,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{1,11} & \overline{G}_{1,12} \\ \overline{G}_{1,21} & \overline{G}_{1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \cdots & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots
\end{array}\right]$$

$$\cdot \begin{bmatrix} h_0 & h_1 & \cdots & | & h_{-1} & h_{-2} & \cdots \end{bmatrix}^T$$

The next operation to finally compute the first term in Equation 3.22 is to find the matrix representation of the term $\begin{pmatrix} I_N & 0 \\ 0 & \bar{\chi} \end{pmatrix}$ and then use it to premultiply $CT_{M_3}$. This premultiplication term performs the operation of the left shift operator $\bar{\chi}$ acting on the second row of the symbol $\begin{pmatrix} \omega A & \chi\bar{a} \\ \bar{\chi}a^T & 0 \end{pmatrix}$ and leaves the elements of the first row in the same positions. Thus the first term $Q_1$ of Equation 3.22 can be finally established in the matrix form

$$\begin{bmatrix}
\begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{1,21} & \overline{G}_{1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-1,11} & \overline{G}_{-1,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-2,11} & \overline{G}_{-2,12} \\ \overline{G}_{-1,21} & \overline{G}_{-1,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \cdots & | & \vdots & \vdots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \ddots & | & \vdots & \vdots & \vdots \\
- & - & - & - & - & + & - & - & - \\
\begin{bmatrix} \overline{G}_{1,11} & \overline{G}_{1,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{-1,21} & \overline{G}_{-1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{-1,11} & \overline{G}_{-1,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\
\begin{bmatrix} \overline{G}_{2,11} & \overline{G}_{2,12} \\ \overline{G}_{1,21} & \overline{G}_{1,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{1,11} & \overline{G}_{1,12} \\ \overline{G}_{0,21} & \overline{G}_{0,22} \end{bmatrix} & \begin{bmatrix} \overline{G}_{0,11} & \overline{G}_{0,12} \\ \overline{G}_{-1,21} & \overline{G}_{-1,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \cdots & | & \vdots & \vdots & \ddots
\end{bmatrix}$$

### 3.2.2.2 Matrix Representation of the Shifted Hankel Operator

To complete the matrix representation of Equation 3.22, we now require the representation of the second term containing the Hankel operator. The infinite matrix representation of the Hankel operator acting only on the negative index Fourier coefficients in the Hankel operator equation $x = H_{M_2}h$ is given by

$$\begin{bmatrix} x_{-1} \\ x_{-2} \\ x_{-3} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} G_{-1} & G_{-2} & G_{-3} & \cdots & \cdots \\ G_{-2} & G_{-3} & G_{-4} & \cdots & \cdots \\ G_{-3} & G_{-4} & G_{-5} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} h_{-1} \\ h_{-2} \\ h_{-3} \\ \vdots \\ \vdots \end{bmatrix}$$

where

$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \end{bmatrix}$ are the Fourier coefficients of $x$,

$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ \vdots \end{bmatrix}$ are the Fourier coefficients of $h$,

and $\begin{bmatrix} G_{-1} & G_{-2} & G_{-3} & \cdots & \cdots \\ G_{-2} & G_{-3} & G_{-4} & \cdots & \cdots \\ G_{-3} & G_{-4} & G_{-5} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$ are the Fourier coefficients of matrix representation of what is termed the block shifted *Hankel operator* with symbol $M_2$

Again, this is just a part of the full form of the Hankel operator. The full representation of the Hankel operator equation $x = H_{M_2} h$ which includes the positive index Fourier coefficients and with the symbol $M_2 = \begin{pmatrix} \bar{\chi}\omega B & a \\ a^T & 0 \end{pmatrix}$ has the form

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ - \\ x_{-1} \\ x_{-2} \\ \vdots \\ \vdots \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots & | & 0 & 0 & \cdots \\ - & - & - & - & - & + & - & - & - \\ \begin{bmatrix} G_{-1,11} & G_{-1,12} \\ G_{-1,21} & G_{-1,22} \\ G_{-2,11} & G_{-2,12} \\ G_{-2,21} & G_{-2,22} \\ G_{-3,11} & G_{-3,12} \\ G_{-3,21} & G_{-3,22} \end{bmatrix} & \begin{bmatrix} G_{-2,11} & G_{-2,12} \\ G_{-2,21} & G_{-2,22} \\ G_{-3,11} & G_{-3,12} \\ G_{-3,21} & G_{-3,22} \\ G_{-4,11} & G_{-4,12} \\ G_{-4,21} & G_{-4,22} \end{bmatrix} & \begin{bmatrix} G_{-3,11} & G_{-3,12} \\ G_{-3,21} & G_{-3,22} \\ G_{-4,11} & G_{-4,12} \\ G_{-4,21} & G_{-4,22} \\ G_{-5,11} & G_{-5,12} \\ G_{-5,21} & G_{-5,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots & | & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & | & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\cdot \begin{bmatrix} h_0 & h_1 & \cdots & | & h_{-1} & h_{-2} & \cdots \end{bmatrix}^T$$

where $G_{-1,11}, G_{-2,11}, \cdots$ are the negative Fourier coefficients of the function $\bar{\chi}\omega B$ in $M_2$,
$G_{-1,12}, G_{-2,12}, \cdots$ are the negative Fourier coefficients of the function $a$ in $M_2$,
$G_{-1,21}, G_{-2,21}, \cdots$ are the negative Fourier coefficients of the function $a$ in $M_2$,
and $G_{-1,22}, G_{-2,22}, \cdots$ are the ( all zero ) negative Fourier coefficients of the function 0 in $M_2$ .

The final version of the second term of Equation 3.22, $\begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix} \mathcal{H}_{M_2}$, is now relatively easy to compute. The matrix $\begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix}$ in front of the Hankel operator is the operation that keeps the second row of $M_2$ in the same positions and shifts the first row of $M_2$ to one position in the right direction. The second term of Equation 3.22 is thus finalised in the Hankel matrix equation $Q_2 = \begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix} \mathcal{H}_{M_2} h$ as

$$\begin{bmatrix} Q_{2,0} \\ Q_{2,1} \\ \vdots \\ \vdots \\ - \\ Q_{2,-1} \\ Q_{2,-2} \\ \vdots \\ \vdots \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & \cdots & | & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots & | & \vdots & \vdots & \ddots \\ - & - & - & - & - & + & - & - & - \\ \begin{bmatrix} G_{-2,11} & G_{-2,12} \\ G_{-1,21} & G_{-1,22} \end{bmatrix} & \begin{bmatrix} G_{-3,11} & G_{-3,12} \\ G_{-2,21} & G_{-2,22} \end{bmatrix} & \begin{bmatrix} G_{-4,11} & G_{-4,12} \\ G_{-3,21} & G_{-3,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\ \begin{bmatrix} G_{-3,11} & G_{-3,12} \\ G_{-2,21} & G_{-2,22} \end{bmatrix} & \begin{bmatrix} G_{-4,11} & G_{-4,12} \\ G_{-3,21} & G_{-3,22} \end{bmatrix} & \begin{bmatrix} G_{-5,11} & G_{-5,12} \\ G_{-4,21} & G_{-4,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\ \begin{bmatrix} G_{-4,11} & G_{-4,12} \\ G_{-3,21} & G_{-3,22} \end{bmatrix} & \begin{bmatrix} G_{-5,11} & G_{-5,12} \\ G_{-4,21} & G_{-4,22} \end{bmatrix} & \begin{bmatrix} G_{-6,11} & G_{-6,12} \\ G_{-5,21} & G_{-5,22} \end{bmatrix} & \cdots & \cdots & | & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \cdots & | & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \ddots & | & \vdots & \vdots & \vdots \end{bmatrix}$$

$$\cdot \begin{bmatrix} h_0 & h_1 & \cdots & | & h_{-1} & h_{-2} & \cdots \end{bmatrix}^T$$

#### 3.2.2.3 Inversion of the Conjugate Toeplitz plus Hankel operator

As shown in Equation 3.22, the update increment function is determined by the inverse of the Jacobian matrix $T_{f,\beta}$. Although a specific algorithm to invert the sum of conjugate Toeplitz plus Hankel matrix has been developed by A.H. Sayed et. al. [39, 79], in this brief study to speed coding, a least square inversion method is adopted using the standard Matlab function `lsqr`.

## 3.3 Conclusions

- In this chapter, the Newton Iteration method is summarized. The NI method is based on the solution to the optimality conditions using operators. Unlike the DI method, there is no approximation of the objective function to a specific form in the NI method. Furthermore, the NI method has a second order convergence rate in theory. Prior to this thesis, no implementation of the NI method was published in any programming language. The chapter details with the computation of the operators so that the use of the NI method is possible.

- In terms of the optimality conditions to the $OPTe$ problem, the operator equation derived by Helton et al. [39] is described for the implementation of the NI method. The optimality conditions for the $OPT_1$ problem and the $OPT_\infty$ problem are also described. The similarity between the conditions for the two problems is seen. This provides the possibility to accelerate the algorithm to solve the $OPT_\infty$ problem.

- The Newton Iteration method formulated in Equation 3.11 updates the solution in a full step length. To compute the update increment function $h$, the Jacobian operator $T'_{f,\beta}$ is required to compute. According to the work by Helton et al. [39], $T'_{f,\beta}$ can be written in Equation 3.22. This chapter gives the matrix representation of $T'_{f,\beta}$ which is described in terms of the Toeplitz operator and the Hankel operator. The arrangement of the Toeplitz operator and the Hankel operator with their symbols is discussed by means of the corresponding Fourier coefficients in the program code. The matrix representation of the Jacobian operator is calculated by summing up the Toeplitz operator and Hankel operator product. It is then possible to compute the update increment function $h$ by inverting the Jacobian operator by a standard matrix inversion algorithm.

# Chapter 4

# Comparison of Algorithms with Numerical Examples

In this section, three examples are used to compare the results from the Matlab Newton Iteration (NI) implementation and Matlab Disk Iteration ( DI ) implementation. The Matlab implementation of the DI method is coded by line-by-line translation from the original ANOPT package in Mathematica [41]. The Matlab code for the Newton iteration method are based on the matrix representation of the operators outlined in the previous chapter.

## 4.1 Example 1

The example appears in the original Mathematica ANOPT codes [41] where the objective function is the scalar function on the unit disk :

$$\Gamma(z, f) = \left\| 0.8 + \left( \frac{1}{z} + f \right)^2 \right\|^2$$

where $f \in \mathbb{H}^\infty$ is the function to be optimized such that

$$\gamma = \sup_{z \in \partial D} \Gamma(z, f)$$

attains its minimum at $\gamma^*$.

Table 4.1.1 and Figure 4.1 shows the solutions computed by the Matlab implementations of the two algorithms.

| | $\gamma^*$ | Iterations | Optimality Test | |
|---|---|---|---|---|
| | | | Flatness | Gradient alignment |
| DI method | 1.0000 | 4 | 1.6301E-7 | 1.9939E-14 |
| NI method | 1.0000 | 4 | 9.2075E-7 | 1.9128E-14 |

Table 4.1.1: Results of Example 1 (256 points and the tolerance = 1E-6)

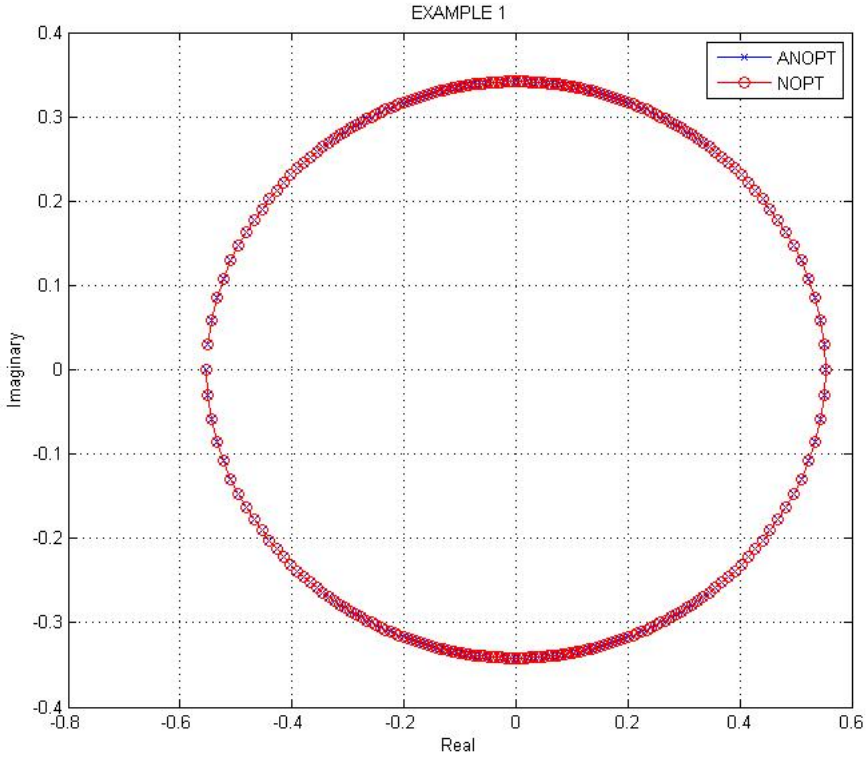



Figure 4.1: Comparison of the solutions by different algorithms (Example1)

## 4.2 Example 2

This two-dimensional $N = 2$ example appears in the original Mathematica ANOPT codes [41] where the objective function is

$$\begin{aligned}\Gamma(z, f_1, f_2) &= \Re\left(\frac{1}{z} + f_1\right)^2 + 4 \cdot \Im\left(\frac{1}{z} + f_1\right)^2 \\ &\quad + \Re\left(\frac{1}{z} + f_2\right)^2 + 0.3 \cdot \Im\left(\frac{1}{z} + f_2\right)^2\end{aligned}$$

where $f_1, f_2 \in \mathbb{H}^\infty$ are the functions to be found and $\Re$ and $\Im$ represent the real part and imaginary part respectively.

Table 4.2.1 shows the results and Figure 4.2 shows the solutions by the Matlab implementation of the two algorithms.

| | $\gamma^*$ | Iterations | Optimality Test | |
|---|---|---|---|---|
| | | | Flatness | Gradient alignment |
| DI method | 2.1469 | 4 | 3.3785E-6 | 0.0086 |
| NI method | 2.1342 | 34 | 0.0096 | 0.1013 |

Table 4.2.1: Results of Example 2 (256 points and the tolerance = 1E-4)

## 4.3 Example 3

This example appears in [39] P.861 where the optimization problem is to find the pair of optimal functions $(f_1, f_2)$

$$\gamma^* = \min_{f_1, f_2 \in \mathbb{H}^\infty} \sup_{z \in \partial D} \Gamma(z, f_1, f_2)$$

where

$$\begin{aligned} \Gamma(z, f_1, f_2) &= \|f_1\|^2 + \|f_2\|^2 + \\ &\quad \|100 + z \cdot f_1 + 0.1 \cdot (f_1 + f_2 + f_1 \cdot f_2)\|^2 + \\ &\quad \|100 + z \cdot f_1 + 0.1 \cdot (f_1 + f_2 + f_1 \cdot f_2)\|^2 \end{aligned}$$

and $f_1, f_2 \in \mathbb{H}^\infty$

Table 4.3.1 shows the results and Figure 4.3 shows the solutions computed by the Matlab implementations of the two algorithms.

| Method | $\gamma^*$ | Iterations | Computing time | Optimality Test | |
|---|---|---|---|---|---|
| | | | | Flatness | Gradient alignment |
| DI | 3800 | 11 | 379.8594 sec | 1.9512E-9 | 8.6135E-6 |
| NI | 3800 | 8 | 32.1590 sec | 4.3480E-12 | 1.0745E-5 |

Table 4.3.1: Results of Example 3 (256 points and the tolerance = 1E-6)

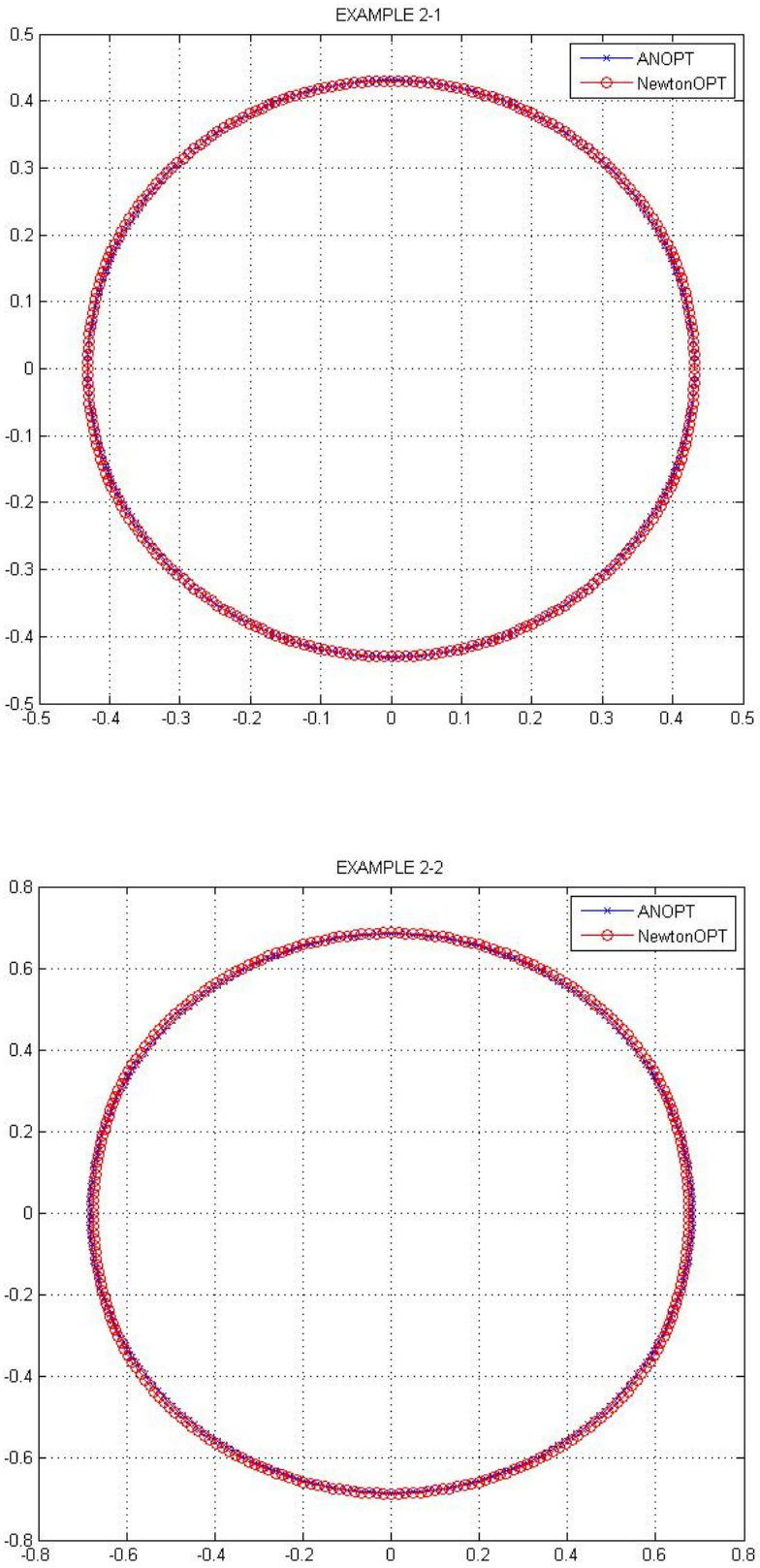



Figure 4.2: Comparison of the solutions by different algorithms (Example2)

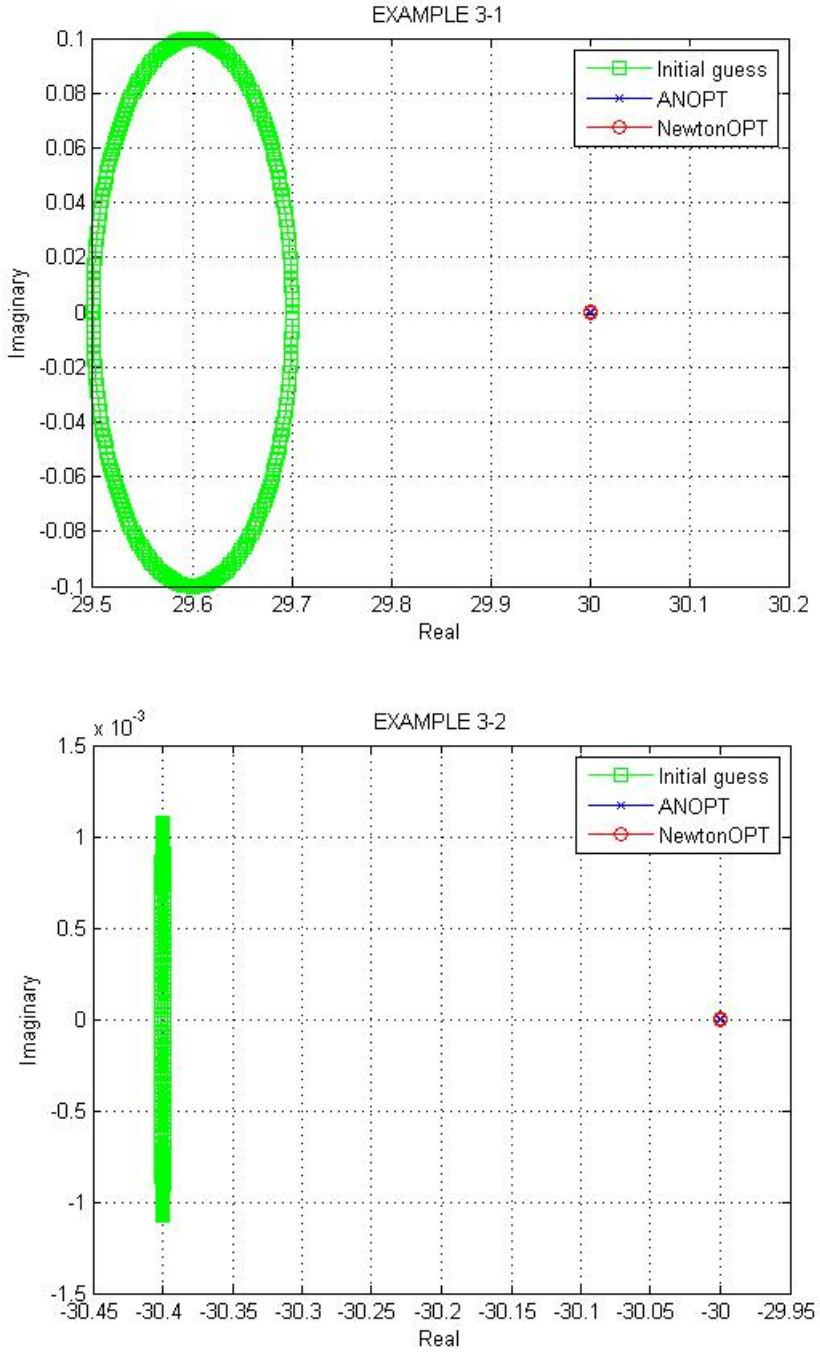



Figure 4.3: Comparison of the solutions by different algorithms (Example3)

## 4.4 Discussions

1. *Example 1*

   The solutions obtained from applying the Matlab implementations of the DI method and NI method are very close to each other. The solutions both converge in 4 iterations. It can be observed that, as we decrease the tolerance down to $10^{-8}$ in order to improve the accuracy, the NI method requires the same number of iterations to converge within the tolerance, as shown in Table 4.4.1.

| | $\gamma^*$ | Iterations | Optimality Test | |
|---|---|---|---|---|
| | | | Flatness | Gradient alignment |
| Disk iteration | 1.0000 | 5 | 1.6710E-9 | 1.9249E-14 |
| Newton iteration | 1.0000 | 5 | 2.4387E-8 | 1.9276E-14 |

Table 4.4.1: Results of Example 1 (256 points and the tolerance = 1E-8)

   It is concluded that the NI method implementation converges to the same solution computed by the implementation of the DI method. In this dimension $N = 1$ problem, the two implementations appear to have similar convergence properties.

2. *Example 2*

   In Figure 4.2, the two solutions can be seen to be close to each other. Both algorithms converge to the same solution. However, as shown in Table 4.2.1, the NI implementation iterated 34 times to converge to the solution while the DI implementation needed only 4 iterations. This may be because of the improvement made in [46] for accelerating the NI method in such optimization problem. Nevetheless, in examples where the variables are more iteractive, it may be expected that the truncation of the second order term in the DI method may then result in more iterations. This conjecture is supported by the results found in Example 3.

3. *Example 3*

   By using the given initial points at $\left(29.6 + 0.1e^{j\theta}, -30.4 - 0.0001e^{j\theta} + 0.001(e^{j\theta})^2\right)$, the optimal solution is algebraically calculated as $(30, -30)$. The results in Figure 4.3 show that the NI implementation converges in 8 iterations whereas the DI method produces the optimal point $(30, -30)$ in 11 iterations. This supports the conclusion that the NI method has a higher ( quadratic ) convergence rate when the dimension is greater than 1 [39].

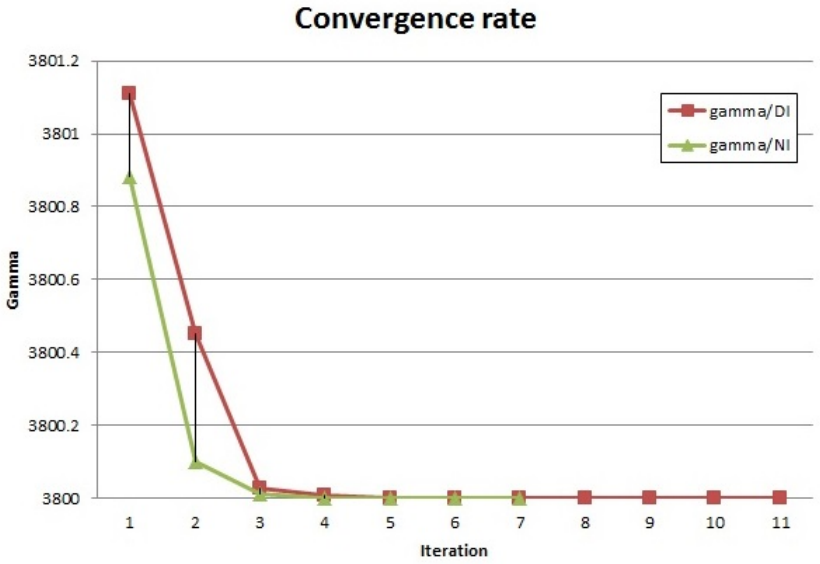


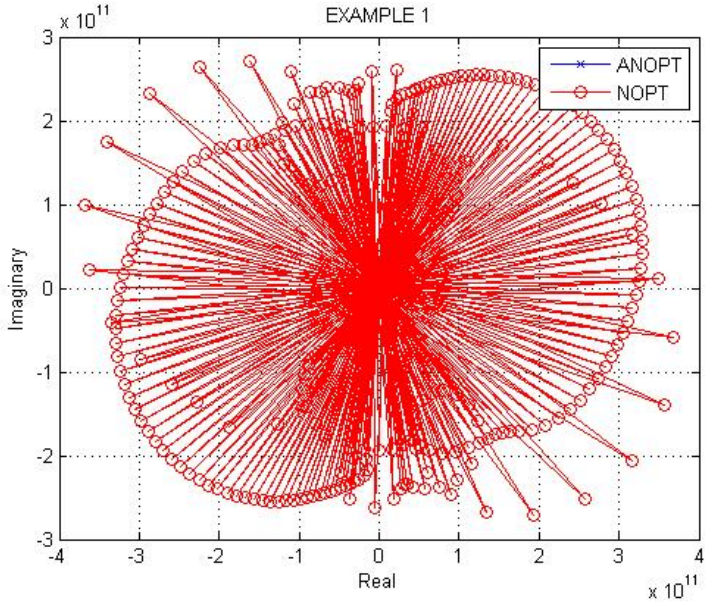

Figure 4.4: Convergence rate of Example 3

| Iteration | Gamma / DI | Gamma / NI |
|---|---|---|
| 1 | 3801.108728 | 3800.8835835 |
| 2 | 3800.450219 | 3800.099895 |
| 3 | 3800.027523 | 3800.011241 |
| 4 | 3800.009851 | 3800.000971 |
| 5 | 3800.000366 | 3800.000061 |
| 6 | 3800.000178 | 3800.000003 |
| 7 | 3800.000138 | 3800.000001 |
| 8 | 3800.000019 | 3800.000000 |
| 9 | 3800.000009 | |
| 10 | 3800.000004 | |
| 11 | 3800.000003 | |

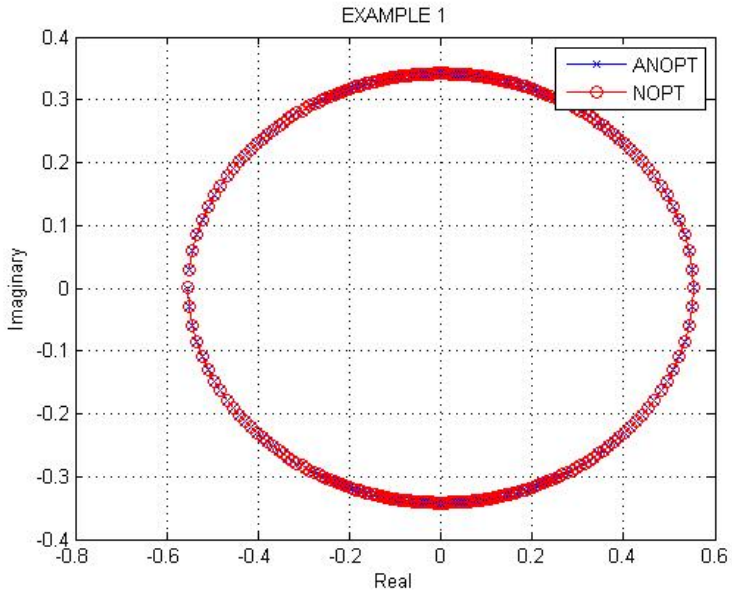
Table 4.4.2: Numerical results in Example 3

## 4.5 Sensitivity of Newton method to the initial guess

The convergence of solutions by the NI implementation is experimentally found to be very sensitive to the initial conditions of the optimisation problem. With initial points away from the optimum, the NI implementation is found to converge to other solutions or even diverge. The behaviour can also be found in the previous three examples as shown in Figure 4.5, Figure 4.6 and Figure 4.7.




(a) Deviation of the solutions in EX1 with the starting points 0.5




(b) Deviation of the solutions in EX1 with the starting points 0.2

Figure 4.5: Deviation of the solutions in EX1 from different starting points

Figure 4.5 shows that in this example if the initial starting points are in the optimal solution circle (e.g. Figure 4.5.b), the solutions are able to converge to the same points. However, if the starting points are ouside the solution circle or near the solution circle, the NI implementation becomes divergent (Figure 4.5.a).

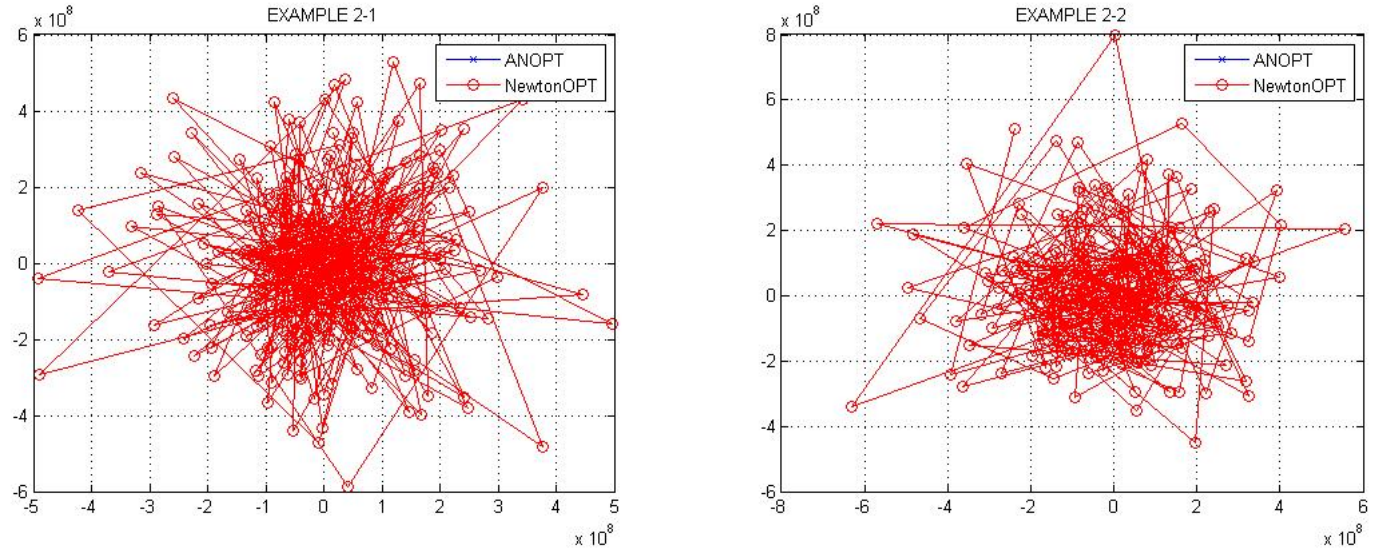
It is also observed in Example 2 that the solutions are very sensitive to the initial points. Figure 4.6.a shows the divergent solution from the point 0.1 and Figure 4.6.b the convergence of the solution from the point 0.001.

In Figure 4.7, the NI implementation is seen to be very sensitive to the initial points. The closeness of the initial points to the optimum determines not only the number of iterations required but also the nature of convergence.

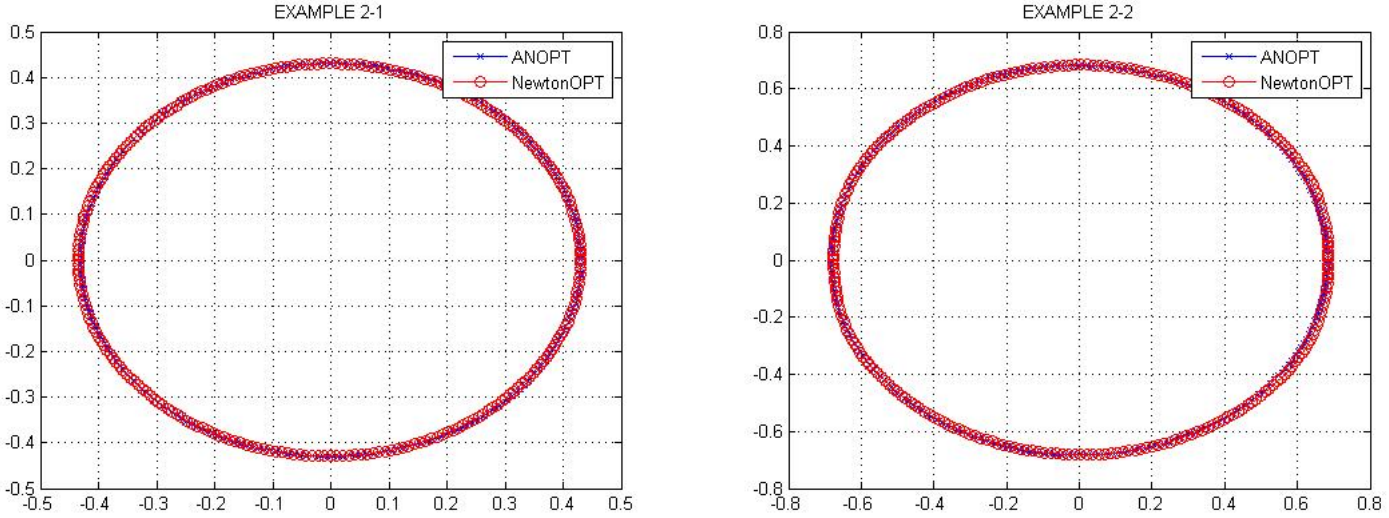
## 4.6 Conclusions

- The comparison of the results of the two algorithms are presented using examples in this chapter. If the objective function contains only one variable, it is found that similar performance in terms of the computing time and the number of iterations used is observed in both methods. However, if the objective function has more than one variable, due to the nature of the second order convergence in the NI method, the NI method converges faster than in the DI method.

- However, it is also experimentally found that the NI method is very sensitive to the initial guess. If the initial points are not close to the optimum, the solution may not be convergent.

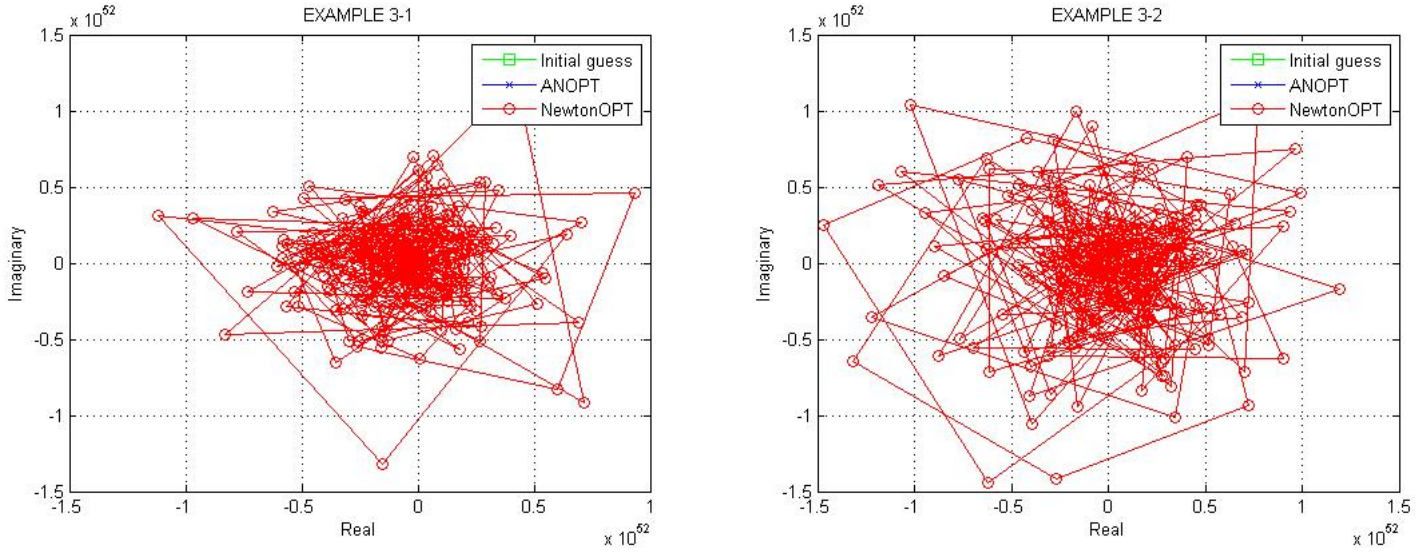



(a) Deviation of the solutions in EX2 with the starting points $(0.1, 0.1)$

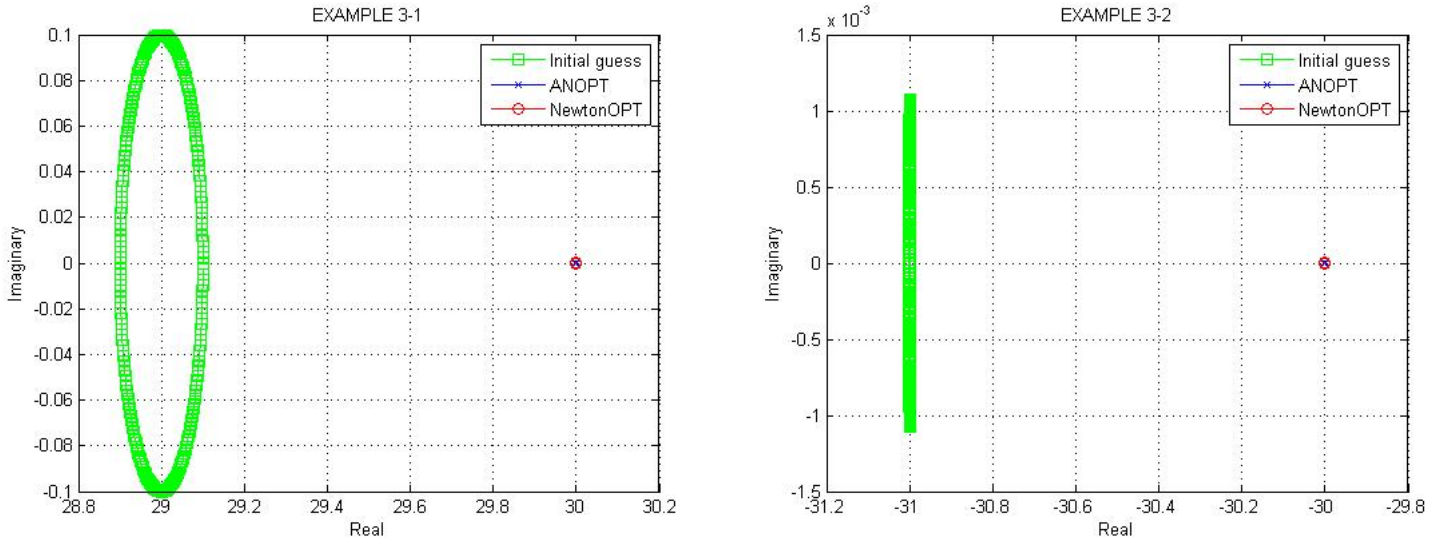



(b) Deviation of the solutions in EX2 with the starting points $(0.001, 0.001)$

Figure 4.6: Deviation of the solutions in EX2 from different starting points




(a) Deviation of the solutions in EX3 with the starting points $\left(25.6+0.1e^{i\theta}, -35.4-0.0001e^{i\theta}+0.001\left(e^{i\theta}\right)^2\right)$




(b) Deviation of the solutions in EX3 with the starting points $\left(29+0.1e^{i\theta}, -31-0.0001e^{i\theta}+0.001\left(e^{i\theta}\right)^2\right)$

Figure 4.7: Deviation of the solutions in EX3 from different starting points

# Chapter 5

# Linear Programming Method for $\mathbb{H}^\infty$ Optimization

## 5.1 Introduction

In this chapter, an approach to solve the $\mathbb{H}^\infty$ control problem in terms of Linear Programming (LP) method by Streit's algorithm [86] is presented. In contrast to the Disk Iteration (DI) method and Newton Iteration (NI) method discussed in Chapter 2 and 3, it is stated [40] that the LP method is capable of dealing with the contraints in both frequency domain and time domain. In Section 5.2, the $\mathbb{H}^\infty$ problem in terms of the Frobenius norm is solved by the LP method. The linear programming problem in terms of Streit's algorithm [86] is summarized by means of the program codes in Section 5.3. An example is illustrated in Section 5.4 and the extension of the linear programming method to MIMO systems is also discussed in this chapter.

## 5.2 Mixed Sensitivity $\mathbb{H}^\infty$-Frobenius Norm Control Problem

### 5.2.1 Internal Stability

In this section, the internal stability requirement of the system is discussed. It is known [109] that the closed-loop system shown in Figure 1.2 is internally stable if the following requirements are satisfied :

1. The complimentary sensitivity function $T$ is stable. This implies that the primary sensitivity function $S$ is also stable.

2. No unstable pole of the plant $G$ is cancelled by any unstable zero of the compensator $K$.

3. No unstable zero of the plant $G$ is cancelled by any unstable pole of the compensator $K$.

To ensure the satisfaction of the internal stability conditions, Helton [47] proposed the interpolation method for $T$ as

$$T(s) = A(s) + B(s) \cdot T_0(s) \tag{5.1}$$

where $s$ represents the Laplace variable, $T_0$ is any continuous function in $\mathbb{H}^\infty$ and $A$, $B$ are the interpolation functions to meet the interpolation conditions in Equation 5.1 of the internal stability criterion :

$$A(p_x) = 1,\ A(z_y) = 0,\ B(p_x) = 0,\ B(z_y) = 0$$

where $p_x$ is the $x$-th RHP pole and $z_y$ is the $y$-th RHP zero of the plant $G$ respectively.

This interpolation method for M.I.M.O. systems is also derived by Zhao [106] as

$$T(s) = A(s) + \alpha(s) \cdot T_0(s) \tag{5.2}$$

where $T_0(s) \in \mathbb{RH}^\infty$ is a continuous matrix-valued function, $\alpha$ is the scalar interpolant funtion defined as $\alpha = \frac{(s-p_x)\cdot(s-z_y)}{(s+p_x)^m(s+z_y)^n}$ where $p_x$ and $z_y$ for $x = 1, \cdots, m$ and $y = 1, \cdots, n$ are the RHP poles and zeros of the plant $G$, and $A$ is the matrix-valued interpolation function that meets the interpolation conditions : $A(p_x) = I$ and $A(z_y) = 0$

In many engineering applications, plants are typically stable. The internal stability requirements of a closed-loop control system in Figure 1.2 with a stable plant depend on the stabilities of the sensitivity functions $S$, $T$, $Q$ and $V$. It is simply proved [106] that for a stable plant $G$, if the function $Q = K[I + KG]^{-1}$ is stable, the complimentary sensitivity function $T = KG[I + KG]^{-1} = GQ$ is stable. We immediately know that the primary sensitivity function $S = I - T$ is also stable. In addition, the function $V$ is stable due to $G$ being stable. If all the four sensitivity functions are stable, the controlled system is internally stable. As a result, the stability of the sensitivity function $Q$ is sometimes used for meeting the stability requirements. This is the well-known *Q-parameterization* method proposed by Zames [104]. In contrast to the interpolation method, this method is used in the rest of the thesis to parameterize $Q$ is potentially attractive for nonparametric control because the computation of the poles and zeros of the plant $G$ is avoided.

### 5.2.2 Mixed Sensitivity $\mathbb{H}^\infty$ Control Problem

It is known [83] that the standard mixed sensitivity control problem is

$$\left\| \begin{bmatrix} W_S S \\ W_T T \end{bmatrix} \right\|_\infty \leq 1 \tag{5.3}$$

where $W_S$ and $W_T$ are the weighting functions for the sensitivity functions $S$ and $T$.

It is suggested [106] that, by means of the Frobenius norm, the Inequality 5.3 can also be approximated as

$$\left\| |W_S S|^2 + |W_T T|^2 \right\|_\infty \leq \frac{1}{2} \tag{5.4}$$

The Inquality 5.4 is then relaxed by choosing the alternative weighting functions $W_S$ and $W_T$ such that

$$\left\| \left| W_S^{'} S \right|^2 + \left| W_T^{'} T \right|^2 \right\|_\infty \leq 1 \tag{5.5}$$

In summary, the optimization problem of the mixed sensitivity control problem can be outlined as

$$\gamma^* = \min_{\omega} \left\| |W_S(j\omega) S(j\omega)|^2 + |W_T(j\omega) T(j\omega)|^2 \right\|_\infty \leq 1 \tag{5.6}$$

In other words, the $\mathbb{H}^\infty$ mixed sensitivity problem is to find the analytic solutions of the functions $S$ and $T$ in Inequality 5.6. Because the analytic property of functions on the unit disk $D$ is well known, as discussed in Chapter 2, the problem in the Equality 5.6 may be converted from the $j\omega$-axis in the complex plane to the unit disk $D$ by a linear fractional transform to give

$$\gamma^* = \min_{\theta \in D} \left\| \left| W_S\left(e^{j\theta}\right) S\left(e^{j\theta}\right) \right|^2 + \left| W_T\left(e^{j\theta}\right) T\left(e^{j\theta}\right) \right|^2 \right\|_\infty \leq 1$$

Due to the relation between the function $S$ and the function $T$, i.e. $S + T = I$, this results in

$$\gamma^* = \min_{\theta \in D} \left\| \left| W_S\left(e^{j\theta}\right) \left(I - T\left(e^{j\theta}\right)\right) \right|^2 + \left| W_T\left(e^{j\theta}\right) T\left(e^{j\theta}\right) \right|^2 \right\|_\infty \leq 1 \tag{5.7}$$

or, due to $T = GQ$,

$$\gamma^* = \min_{\theta \in D} \left\| \left| W_S\left(e^{j\theta}\right) \left(I - G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right)\right) \right|^2 + \left| W_T\left(e^{j\theta}\right) G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right) \right|^2 \right\|_\infty \leq 1 \tag{5.8}$$

It is mentioned in the previous chapter that if the system is internally stable, all the four sensitivity functions must be stable. As a result, it is possible to reformulate the original optimization problem in Equation 5.7 with the optimal analytic solution $T$ to the one in Equation 5.8 with the analytic function $Q$. That is, for example, the optimization problem in Inequality 5.8 for SISO systems degenerates to the min-max problem

$$\gamma^* = \min_{\theta \in D} \max \left( \left| W_S \left(1 - GQ\right) \right|^2 + \left| W_T \left(GQ\right) \right|^2 \right) \tag{5.9}$$

For MIMO systems, the optimization problem can be related to the Hadamard weighted $\mathbb{H}^\infty$-Frobenius norm problem introduced by van Diggelen and Glover [92]. The significant advantage of using Hadamard weighted $\mathbb{H}^\infty$-Frobenius norm in the mixed sensitivity control problem is that using this norm provides an intuitive and independent way to adjust the elements of the weighting function, which is possible to be used in the decoupling control problem by the $\mathbb{H}^\infty$ method. A systematic controller design procedure based on the loop shaping design procedure [67] in terms of the Hadamard weighted $\mathbb{H}^\infty$-Frobenius norm, defined as $\sup_\omega \|F(j\omega)\| = \sup_\omega \left(trace(F(-j\omega) \cdot F(j\omega))\right)^{1/2}$ where $F(j\omega)$ is the frequency response function, is also proposed in [91]. In this chapter, by using the Hadamard weighted $\mathbb{H}^\infty$-Frobenius norm, we develop a nonparametric frequency-response-based approach only for stable plants. In summary, the optimization problem in Inequality 5.8 for MIMO systems becomes

$$\gamma^* = \min_{\theta \in D} \left\| \left| W_S\left(e^{j\theta}\right) \star \left(I - G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right)\right) \right|^2 + \left| W_T\left(e^{j\theta}\right) \star \left(G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right)\right) \right|^2 \right\|_\infty \tag{5.10}$$

where the Hadamard weighting $\star$ denotes 'element-by-element weighting of a transfer function' [91]. For a $2 \times 2$ M.I.M.O. system, for example, the optimization problem can be written as

$$\begin{aligned} \gamma^* = \min_{\theta \in D} \quad & \left| W_{S,11} \left(1 - G_{11}Q_{11} - G_{12}Q_{21}\right) \right|^2 + \left| W_{T,11} \left(G_{11}Q_{11} + G_{12}Q_{21}\right) \right|^2 + \\ & \left| W_{S,12} \left(0 - G_{12}Q_{22} - G_{11}Q_{12}\right) \right|^2 + \left| W_{T,12} \left(G_{12}Q_{22} + G_{11}Q_{12}\right) \right|^2 + \\ & \left| W_{S,21} \left(0 - G_{21}Q_{11} - G_{22}Q_{21}\right) \right|^2 + \left| W_{T,21} \left(G_{21}Q_{11} + G_{22}Q_{21}\right) \right|^2 + \\ & \left| W_{S,22} \left(1 - G_{22}Q_{22} - G_{21}Q_{12}\right) \right|^2 + \left| W_{T,22} \left(G_{22}Q_{22} + G_{21}Q_{12}\right) \right|^2 \end{aligned} \tag{5.11}$$

$$\tag{5.12}$$

Similar results can thus be obtained for higher dimension MIMO systems. As a result, the optimal stabilizing controller $K^*$ is then calculated by

$$K^* = Q^* \left[ I - GQ^* \right]^{-1} \tag{5.13}$$

It is noted that, at this point, the functions are in the form of the frequency response. To implement the controller in reality, it is normally required to obtain the parametric functions of the controller $K^*$. Finding the corresponding rational function of $K^*$ can be accomplished by using Trefethen's Chebyshev approximation method [90] via Carathéodory-Fejér theorem [16], which is known as CF approximation method [89].

### 5.2.3 Algorithm to Solve the Mixed Sensitivity Problem

Given a stable plant $G$, the controller design method in the Hadamard weighted $\mathbb{H}^\infty$-Frobenius mixed sensitivity norm problem by Streit's algorithm [86] is outlined in follow the steps:

1. Select the weighting functions $W_S$ and $W_T$ that meet the performance specifications

2. Transform the frequency responses of $W_S(j\omega)$ and $W_T(j\omega)$ to those of $W_S\left(e^{j\theta}\right)$ and $W_T\left(e^{j\theta}\right)$ on the unit disk by a linear fractional transform

3. Calculate the sensitivity function $Q$ by solving the Hadamard weighted $\mathbb{H}^\infty$-Frobenius norm optimization problem

$$\begin{aligned} \gamma^* \quad = \quad & \min_{\theta \in D} \Big\| \left| W_T\left(e^{j\theta}\right) \star \left( G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right) \right) \right|^2 + \\ & \left| W_T\left(e^{j\theta}\right) \star \left( G\left(e^{j\theta}\right) Q\left(e^{j\theta}\right) \right) \right|^2 \Big\|_\infty \end{aligned}$$

4. Compute the controller by $K = Q\left[1 - GQ\right]^{-1}$

5. Calculate the rational function of the controller $K$ by using CF approximation method [89] and map the function $K$ from the unit circle domain to the frequency domain by the inverse linear fractional transform to implement the controller.

## 5.3 Linear Programming Method

The Algorithm635 proposed by Streit [87] for solving the systems of complex linear equations in terms of $\mathcal{L}_\infty$ norm, which is summarized below.

Given a $n \times m$ matrix $A \in \mathbb{C}$, a $n \times r$ matrix $B \in \mathbb{C}$, and the vectors $f \in \mathbb{C}^m$, $g \in \mathbb{C}^r$, $a \in \mathbb{C}^n$, $d \in \mathbb{R}^n$ and $c \in \mathbb{R}^r$, the optimization problem

$$\min_{z \in \mathbb{C}} \|zA - f\|_\infty \tag{5.14}$$

subject to

$$\begin{aligned} |z - a| &\leq d \\ |zB - g| &\leq c \end{aligned} \tag{5.15}$$

where $\|\cdot\|_\infty$ denotes the $\mathcal{L}_\infty$ norm, can be discretized to the problem

$$\min_{\varepsilon \in \mathbb{R}, z \in \mathbb{C}^n} \varepsilon \tag{5.16}$$

subject to

$$\begin{aligned} |zA_i - f_i|_D \leq \varepsilon \quad &, \quad i = 1, \cdots, m \\ |zB_i - g_i|_D \leq c_i \quad &, \quad i = 1, \cdots, r \\ |z - a_i|_D \leq d_i \quad &, \quad i = 1, \cdots, n \end{aligned} \tag{5.17}$$

where $A_i$ and $B_i$ are the $i$-th columns of the matrices $A$ and $B$.

This algorithm solving the discretized problem is originally published in [86] and implemented in the FORTRAN language in [87]. Several applications of the Algorithm635 are also studied in [40, 72, 85, 88]. It is seen from the above derivation that the algorithm is based on the extension of the solution to the complex semi-infinite program (SIP) formulation of the unconstrained problem to the constrained optimization problem.

It is shown in [87] that the norm $|u|_\infty$ can be discretised as a $p$-th polygon : $|u|_D = \max_{\theta \in D} \left( u^R \cos\theta + u^I \sin\theta \right)$ where $D = \{\theta_1, \theta_2, \ldots, \theta_p\}$ is the subset of the interval $[0, 2\pi)$ and $u^R$,$u^I$ are the real and imaginary parts of $u$ respectively . The discretised problem 5.16 can then be transformed to the standard constrained linear programming problem [87] :

$$\min_{\begin{bmatrix} z^R & z^I & \varepsilon \end{bmatrix} \in \mathbb{R}} \begin{bmatrix} z^R & z^I & \varepsilon \end{bmatrix} \begin{bmatrix} 0_n & 0_n & 1 \end{bmatrix}^T \tag{5.18}$$

subject to $\varepsilon \geq 0$ and for each $\theta \in [0, 2\pi]$

$$\begin{bmatrix} z^R & z^I & \varepsilon \end{bmatrix} \begin{bmatrix} A^R \cos\theta + A^I \sin\theta & B^R \cos\theta + B^I \sin\theta & I\cos\theta \\ A^R \sin\theta - A^I \cos\theta & B^R \sin\theta - B^I \cos\theta & I\sin\theta \\ -1_m & 0_r & 0_n \end{bmatrix}$$
$$\leq \begin{bmatrix} f^R \cos\theta + f^I \sin\theta & c + g^R \cos\theta + g^I \sin\theta & d + a^R \cos\theta + a^I \sin\theta \end{bmatrix} \tag{5.19}$$

It is reported [15] that the linear programming problem is usually solved by the simplex method. The simplex method, however, requires large memory capacity to store the huge matrices in this problem [87]. A revised simplex method is presented by Streit [87] to reduce the memory storage for solving the corresponding dual problem. In the Matlab programs to implement Streit's algorithm, the interior point method is used by calling the standard code `linprog` [105] for solving the large scale linear programming problem.

It is also noted in [87] that since the discretised problem is an approximation to the original problem, there may be no feasible solution to the original problem but we can still calculate a solution to the discrete problem. It is discussed in [87] that this is because the region of the constraints in the Inequalities 5.17 is larger than those in the Inequalities 5.15 . However, as the discretisation number of $p$ increases, the discrete problem grows closer to the original problem and the *false* solution to the discrete problem can be excluded. In [87], the discretisation number $p$ is suggested to be 1024. However, it is known that increasing the discretisation number $p$ increases the computing efforts dramatically as shown in Figure 5.1. It is observed in the figure that the computing time starts to increase potentially from the point $p = 16$. In the experience of running many similar Matlab programmes, if a feasible solution is found, the value of $p$ is not neccessary to be 1024 and $p = 16$ is enough to provide a satisfactorily accurate solution.

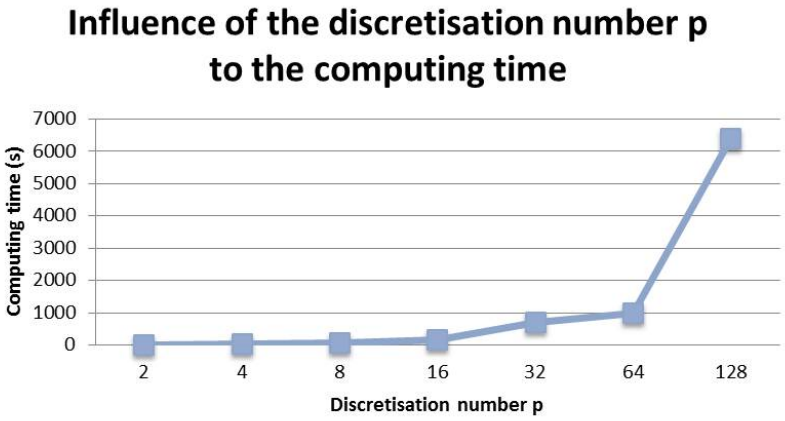



Figure 5.1: The influence of the discretisation number $p$ to the computing time

- **Application of the Linear Programming Method to the Mixed Sensitivity Control Problem**

To solve the mixed sensitivity problem in Equation 5.8, we restate the problem to be a constrained optimization problem as

$$\gamma^* = \min_{\theta \in D} \|W_S \star (1 - GQ)\|_\infty \tag{5.20}$$

subject to

$$\|W_T \star (GQ)\|_\infty \leq 1 \tag{5.21}$$

We first focus on this relaxed mixed sensitivity problem for a SISO system as follows and extend it to MIMO systems afterwards :

$$\gamma^* = \min_{\theta \in D} \|W_S (1 - GQ)\|_\infty \tag{5.22}$$

subject to

$$\|W_T (GQ)\|_\infty \leq 1 \tag{5.23}$$

Due to $Q \in \mathbb{H}^\infty$, the power expasion of $Q$ up to $n$-the order can be expressed in the form

$$
\begin{aligned}
Q\left(e^{i\theta}\right) &= z_0 + z_1 e^{j\theta} + z_2 e^{2j\theta} + \cdots + z_n e^{nj\theta} && (5.24)\\
&= \begin{bmatrix} z_0 & z_1 & z_2 & \cdots & z_n \end{bmatrix} \cdot \begin{bmatrix} e^0 & e^{j\theta} & e^{2j\theta} & \cdots & e^{njn\theta} \end{bmatrix}^T && (5.25)\\
&= \mathbf{z} \cdot \mathbf{A}\left(e^{j\theta}\right) && (5.26)
\end{aligned}
$$

where $\mathbf{z} = \begin{bmatrix} z_0 & z_1 & z_2 & \cdots & z_n \end{bmatrix}$ and $\mathbf{A} = \begin{bmatrix} e^0 & e^{j\theta} & e^{2j\theta} & \cdots & e^{jn\theta} \end{bmatrix}^T$.
Therefore, the mixed sensitivity optimization problem becomes

$$\gamma^* = \min_{\theta \in D} \left\| W_S\left(e^{j\theta}\right)\left(1 - G\left(e^{j\theta}\right)\mathbf{zA}\left(e^{j\theta}\right)\right)\right\|_\infty \tag{5.27}$$

subject to

$$\left\| W_T\left(e^{j\theta}\right)\left(G\left(e^{j\theta}\right)\cdot \mathbf{zA}\left(e^{j\theta}\right)\right)\right\|_\infty \leq 1 \tag{5.28}$$

In the spirit of Streit's algorithm, the problem can then be discretized as

$$\min_{\varepsilon \in \mathbb{R}, z \in \mathbb{C}^n} \varepsilon$$

subject to

$$
\begin{aligned}
\left|\mathbf{z}\tilde{A}_k - W_{S,k}\right|_D &\leq \varepsilon \quad , \quad k = 1, \cdots, n\\
\left|\mathbf{z}\tilde{B}_k - 0\right|_D &\leq 1 \quad , \quad k = 1, \cdots, n
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{A}_k &= W_S\left(e^{jk\theta}\right) G\left(e^{jk\theta}\right) A\left(e^{jk\theta}\right)\\
W_{S,k} &= W_S\left(e^{jk\theta}\right)\\
\tilde{B}_k &= W_T\left(e^{jk\theta}\right) G\left(e^{jk\theta}\right) A\left(e^{jk\theta}\right)
\end{aligned}
$$

and $n \in \mathbb{Z}^+$ is the number of discretizing points. Such problem is closely related to the linear programming problem proposed by Streit [86] and is thus solvable by the standard Matlab algorithm:`linprog` [105].

- **Extension of the Linear Programming Method for MIMO Systems**

The above derivation may also extend for the mixed sensitivity problem for MIMO systems. For instance, for a $2 \times 2$ system, the optimization problem stated in Inequality 5.11 may be re-formulated to the constrained optimization problem :

$$\gamma^* = \min_{\theta \in D} \left\| W_{S,11} \left(1 - G_{11}Q_{11} - G_{12}Q_{21}\right) \right\|_\infty \tag{5.29}$$

subject to

$$\begin{aligned}
\left\| W_{T,11} \left(G_{11}Q_{11} + G_{12}Q_{21}\right) \right\|_\infty &\leq 1 && (5.30)\\
\left\| W_{S,12} \left(0 - G_{12}Q_{22} - G_{11}Q_{12}\right) \right\|_\infty &\leq 1 && (5.31)\\
\left\| W_{T,12} \left(G_{12}Q_{22} + G_{11}Q_{12}\right) \right\|_\infty &\leq 1 && (5.32)\\
\left\| W_{S,21} \left(0 - G_{21}Q_{11} - G_{22}Q_{21}\right) \right\|_\infty &\leq 1 && (5.33)\\
\left\| W_{T,21} \left(G_{21}Q_{11} + G_{22}Q_{21}\right) \right\|_\infty &\leq 1 && (5.34)\\
\left\| W_{S,22} \left(1 - G_{22}Q_{22} - G_{21}Q_{12}\right) \right\|_\infty &\leq 1 && (5.35)\\
\left\| W_{T,22} \left(G_{22}Q_{22} + G_{21}Q_{12}\right) \right\|_\infty &\leq 1 && (5.36)
\end{aligned}$$

In principle, the problem is possibly solved by the LP method in the spirit of the LP method for SISO systems. It may be solvable by other techniques of convex optimization in [14, 15]. In the thesis, the software CVX [33, 34] providing the solution to convex problems will be used in the next chapter to speed up the LP method.

## 5.4 An Application for the Linear Programming Method

In this section, the example in [83] P.274 is used to illustrate the proposed approach by the LP method. The system in the example has the structure of the multiplicative uncertainty with the weighting function $W_I$ as shown in Figure 5.2 where $K$ is the controller to be calculated. The transfer function of the nominal plant $G$ and the weighting function on the uncertainty are given as

$$G = \frac{1}{s+1}$$

$$W_I = \frac{0.125s + 0.25}{0.03125s + 1}$$

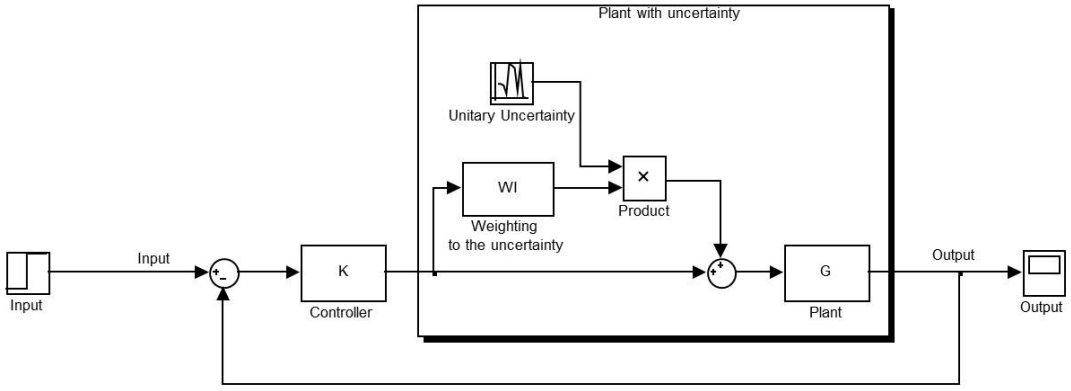



Figure 5.2: System configuration with multiplicative uncertainty where the plant$G = \frac{1}{s+1}$, the weighting function $W_I = \frac{0.125s+0.25}{0.03125s+1}$ and the controller $K$

### 5.4.1 $\mathbb{H}^\infty$ Controller Design using the Parametric Plant

The first illustration of the $\mathbb{H}^\infty$ robust controller design method by linear programming method is presented in terms of the given nominal plant $G$. The control structure used in this section is illustrated in Figure 5.3.

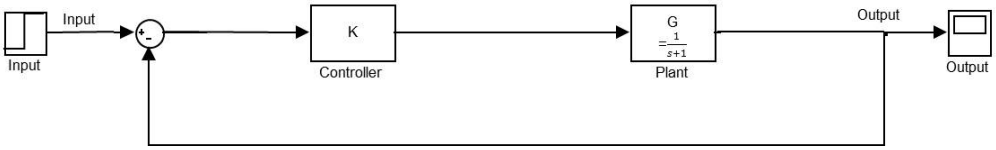



Figure 5.3: Control system based in Figure 5.2 for the parametric controller design method

The frequency responses of $G$ are calculated in the selected frequency range $[0, 325.94]$ Hz with 1024 sampling points. The weighting functions for the functions $S$ and $T$ are selected as

$$W_T = \frac{1.2s + 1}{0.001s + 1.2} \tag{5.37}$$

$$W_S = \frac{s + 1.6}{1.6s + 0.01} \tag{5.38}$$

The mixed sensitivity control problem is then formulated as

$$\gamma^* = \min_{s\in j\omega} |W_S(1-GQ^*)|^2 + |W_T(GQ^*)|^2 \tag{5.39}$$

$$= \min_{s\in j\omega} \left(\frac{s+1.6}{1.6s+0.01}\right)^2 \left|1-\frac{1}{s+1}Q_1^*\right|^2 + \tag{5.40}$$

$$\left(\frac{1.2s+1}{0.001s+1.2}\right)^2 \left|\frac{1}{s+1}Q_1^*\right|^2 \leq 1 \tag{5.41}$$

where $Q_1^*$ is the optimal analytic solution, which can be expanded as $Q_1^*\left(e^{j\theta}\right) = z_0 + z_1 e^{j\theta} + z_2\left(e^{j\theta}\right)^2 + \cdots + z_{n-1}\left(e^{j\theta}\right)^{n-1}$ and $n$ is the highest order of the power expansion. In this example, $n$ is chosen as 60.

The linear programming method is used to solve the optimization problem. It is found that the optimal solution $Q_1^*$ is computed as

$$Q_1^* = \frac{0.0002885s^5 + 0.02432s^4 + 0.6827s^3 + 11.51s^2 + 12.45s + 8.116}{0.0004767s^5 + 0.02964s^4 + 0.7815s^3 + 11.52s^2 + 11.53s + 8.131}$$

and the other performance functions are calculated to be

$$T_1^* = \frac{0.0002885s^5 + 0.02432s^4 + 0.6827s^3 + 11.51s^2 + 12.45s + 8.116}{0.0004767s^6 + 0.03012s^5 + 0.8111s^4 + 12.3s^3 + 23.06s^2 + 19.67s + 8.131}$$

$$S_1^* = \frac{0.0004767s^6 + 0.02983s^5 + 0.7868s^4 + 11.62s^3 + 11.54s^2 + 7.216s + 0.01504}{0.0004767s^6 + 0.03012s^5 + 0.8111s^4 + 12.3s^3 + 23.06s^2 + 19.67s + 8.131}$$

$$V_1^* = \frac{5.323e-06s^{10} + 0.0007357s^9 + 0.04507s^8 + 1.591s^7 + 31.34s^6 + \cdots}{5.527e-06s^{11} + 0.000723s^{10} + 0.04658s^9 + 1.652s^8 + 34.05s^7 + 397s^6 + \cdots}$$
$$\frac{344.7s^5 + 391.1s^4 + 261.3s^3 + 18.66s^2 + 0.4688s + 0.00253}{1108s^5 + 1409s^4 + 981.7s^3 + 335.9s^2 + 20.82s + 0.4511}$$

Observing the four functions, we immediately know that the internal stability requirements are met since the four functions are all stable (i.e. all the poles are in the L.H.P.). The controller is then computed as a fifth-order rational function

$$K_1 = \frac{0.01159s^5 + 0.784s^4 + 28.45s^3 + 35.28s^2 + 2.427s + 0.05548}{0.01845s^5 + 0.9948s^4 + 28.72s^3 + 2.211s^2 + 0.05729s + 0.0003117}$$

In Figure 5.4, it is also seen that the primary sensitivity function $S_1^*$ is bounded by the inverse of the weighting function $W_S$. The same observation can be found in Figure 5.5 where the complementary sensitivity function $T_1^*$ is also bounded by the inverse of the weighting function $W_T$. It is concluded that the controlled

system meets the desired requirement in terms of the functions $S_1^*$ and $T_1^*$.

Futhermore, the performance of the system in time domain can be simulated by observing the step response, whish is illustrated in Figure 5.6. The settling time of this response is about 3.5 seconds and there is no overshoot in the system response.

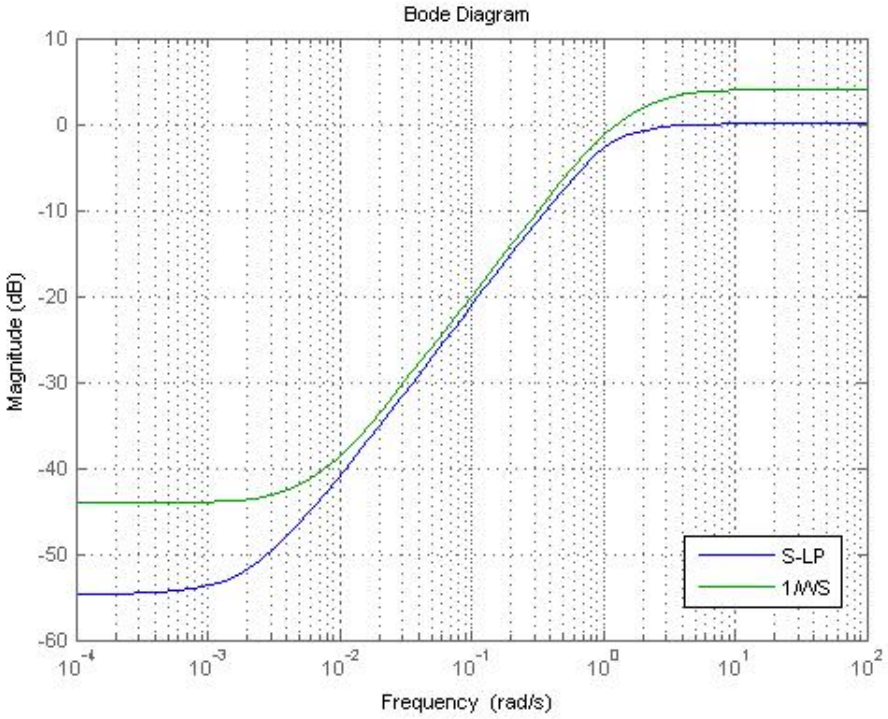



Figure 5.4: Bode diagram of the primary sensitivity function $S_1$ and the weighting function $1/W_S$

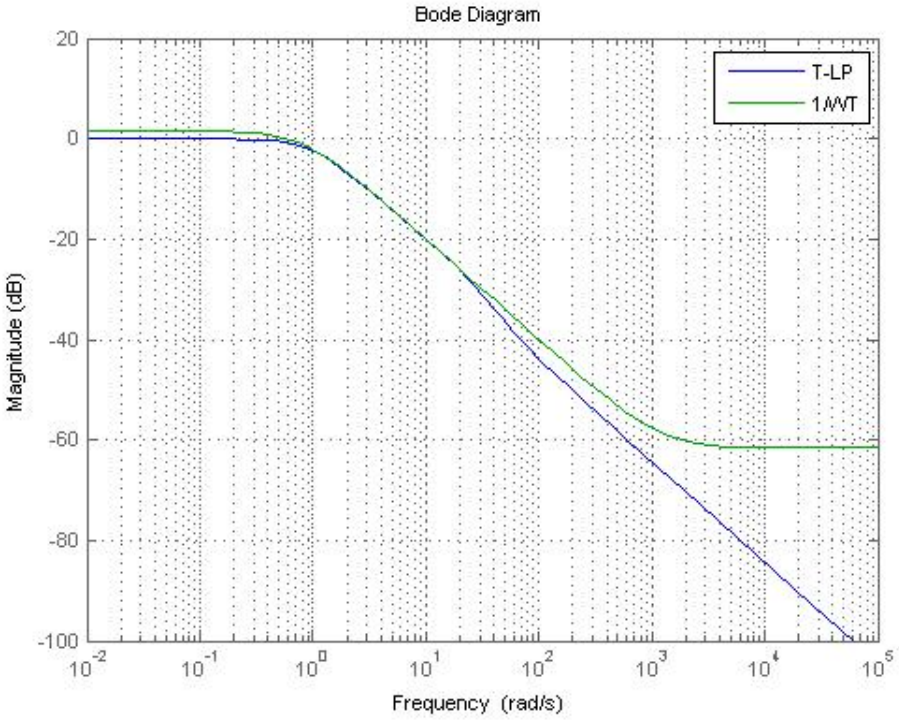



Figure 5.5: Bode diagram of the complementary sensitivity function $T_1$ and the weighting function $1/W_T$

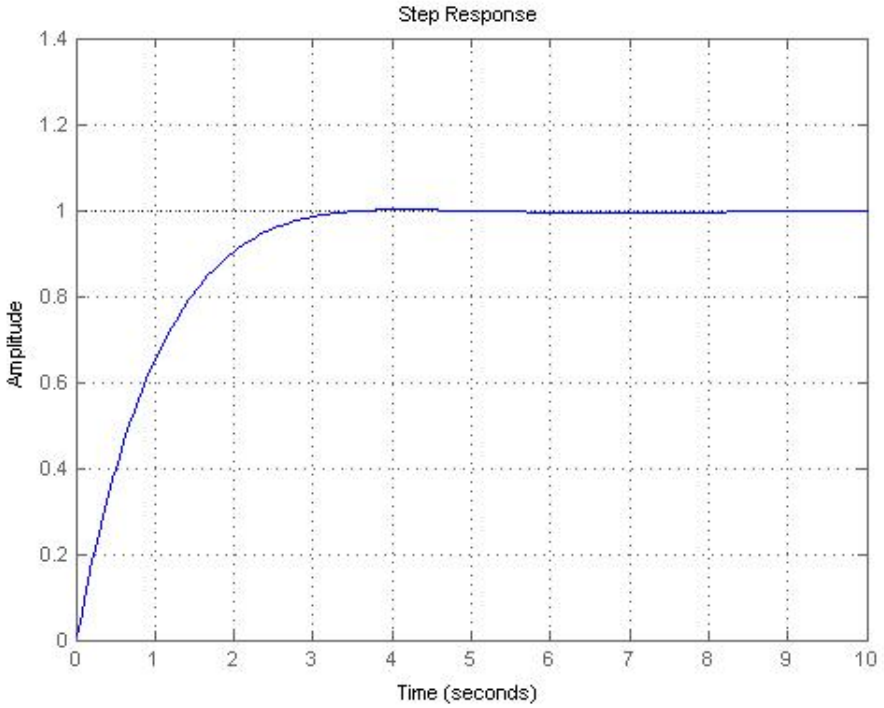



Figure 5.6: Step response simulation of the system in terms of $T_1$

The above simulation results demonstrate the good performance of the controller $K_1$ by the $\mathbb{H}^\infty$ controller design method. However, as there exists the uncertainty in the system as shown in Figure 5.2, it is important to examine whether the controller still has the same performance for the uncertain plant. Figure 5.7 shows the comparison between the measured response for the uncertain plant and the simulated response for the nominal plant $G$. It is observed that the controller has the same performance as expected in the simulation. The figure also shows the controller $K_1$ is robust to the plant uncertainty.

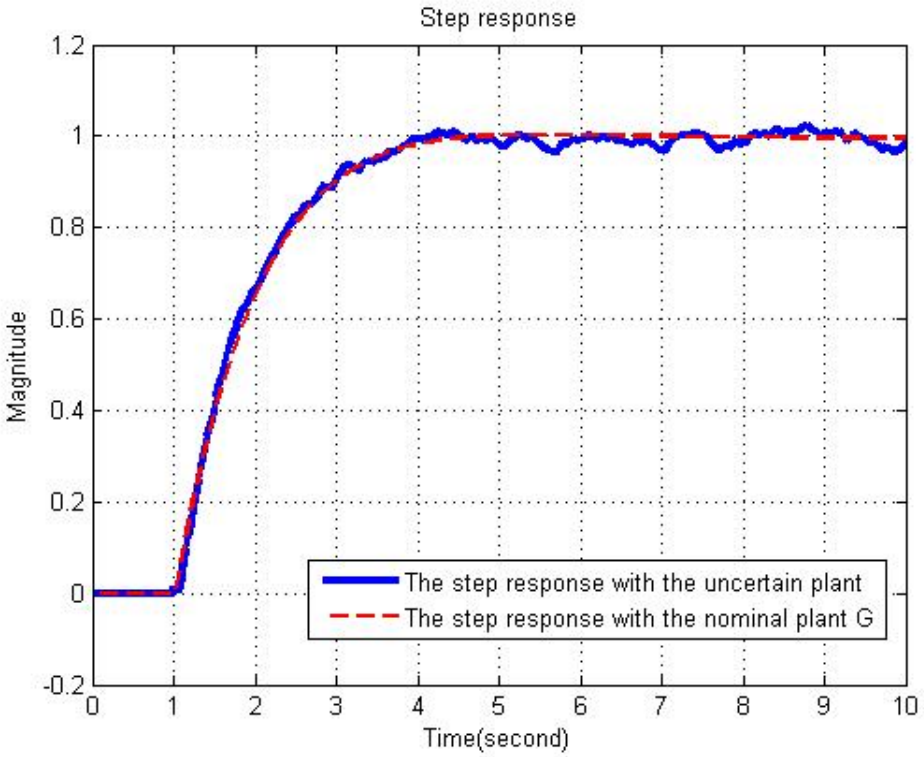



Figure 5.7: Comparison of the measured response (Blue) and the simulated response (Red)

### 5.4.2 Nonparametric $\mathbb{H}^\infty$ Controller Design

In Figure 5.2, there exists multiplicative uncertainties in the system, which normally results in the deviation between the frequency response of the given nominal plant $G$ and the measured frequency response from the input and output of the plant model. Although the uncertainty in the plant model is often treated in terms of the system robustness, in practice, the nominal plant $G$ is usually unknown. However, in the parametric controller design method, the plant model is required to be identified in the first place. This can be accomplished by using several common techniques to obtain the nominal model by using system

identification methods, e.g. disk approximation as shown in Figure 5.8, and the measured frequency response (see [83] for details) etc..

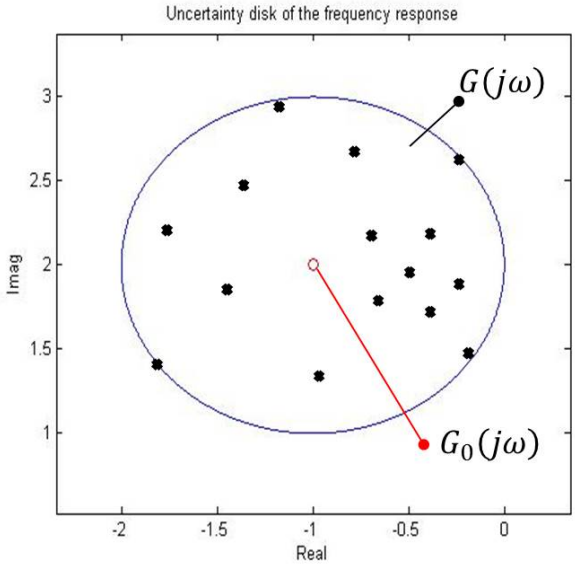



Figure 5.8: The uncertainty disk of the frequency response $G(j\omega)$

In this section, the nonparametric controller design method is adopted in terms of a set of frequency response functions identified from the multiplicative perturbed plant. As the nonparametric method does not require the system identification for the plant model, the method potentially gives more accurate frequency response of the controller and may save much time on system calibration. The nonparametric method contains four steps shown in the following and illustrated in Figure 5.9. The first step is to obtain the frequency response of the plant by Fourier Transform. The second, third, and fourth steps are the same procedure discussed in Section 5.2.3.

### i) Compute the frequency response of the plant

In this example, the simulation to compute the frequency response of the plant model based on the control system in Figure 5.2 is illustated in Figure 5.10. In Figure 5.10, the frequency response is obtained by Fast Fourier Transform (FFT) of the caculation of the output response over the random input signal. The resulting frequency response (by taking the mean at each frequency) is shown in Figure 5.11.

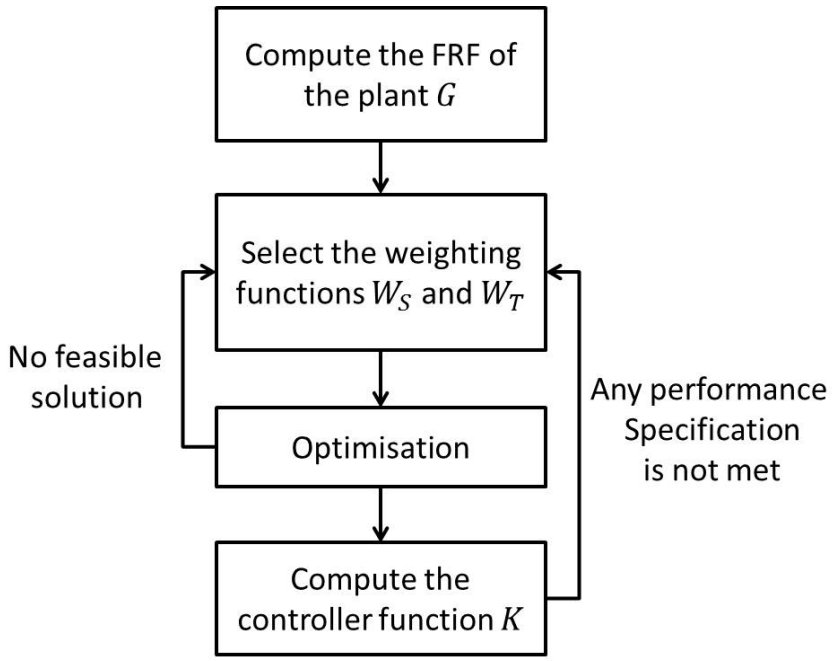



Figure 5.9: The nonparametric controller design method

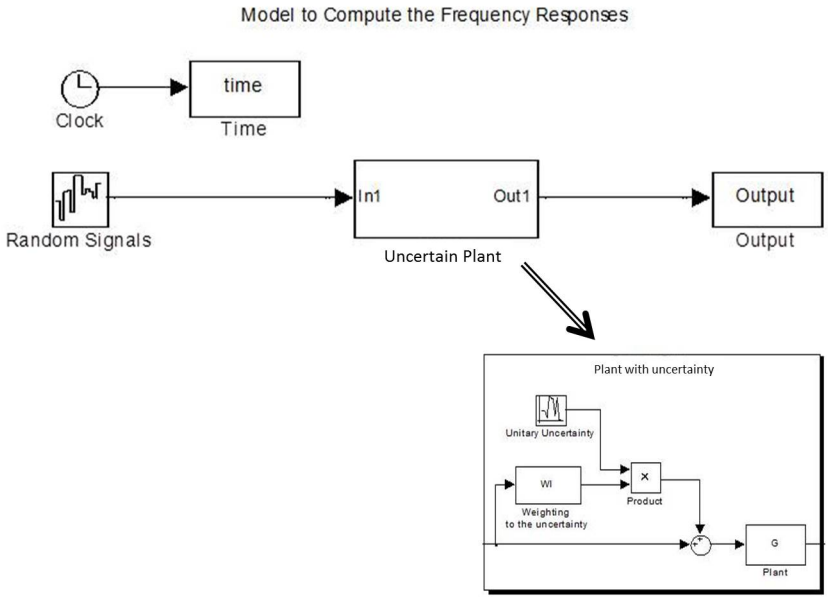



Figure 5.10: Computation of the frequency response of the uncertain plant

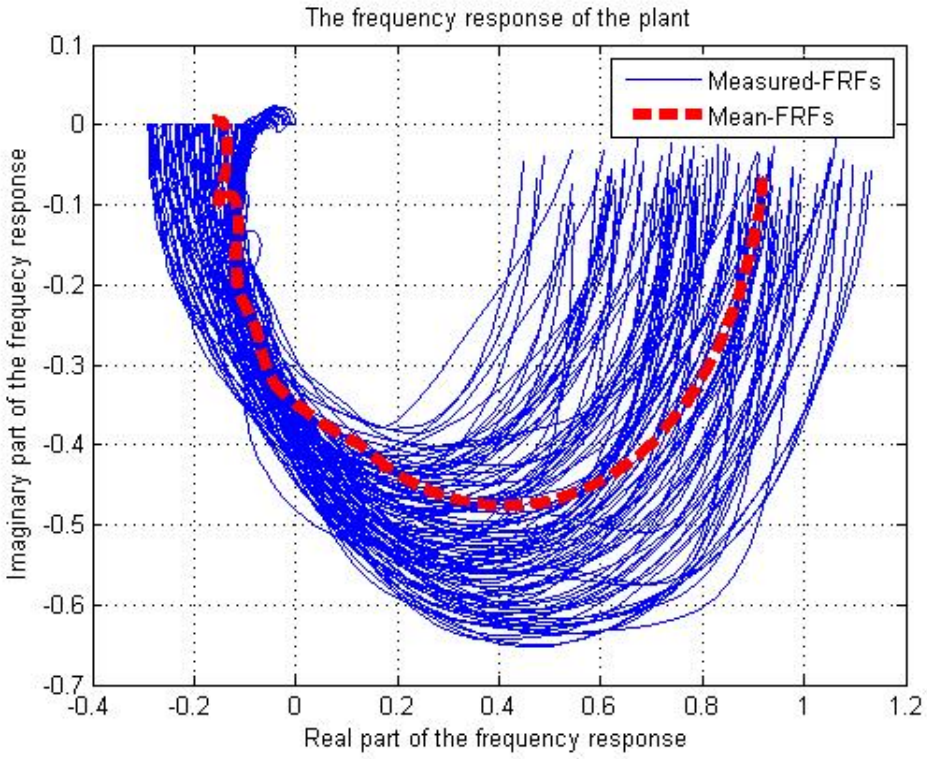



Figure 5.11: Comparison of the measured frequency response (FRF) and the average frequency response (FRF)

### ii) Select the weighting functions $W_T$ and $W_S$

The same weighting functions for the functions $S$ and $T$ used in Equation 5.37 and 5.38 are selected as

$$W_T = \frac{s+1}{0.001s+1.8}$$

$$W_S = \frac{s+1.3}{1.8s+0.01}$$

### iii) Optimisation

Therefore, the $\mathbb{H}^\infty$ mixed sensitivity control problem is formed as

$$\gamma^* = \min_{s\in j\omega} |W_S(1-GQ^*)|^2 + |W_T(GQ^*)|^2 \tag{5.42}$$

$$= \min_{s\in j\omega} \left(\frac{s+1.3}{1.8s+0.01}\right)^2 |1-FRF_P\cdot Q_2^*|^2 + \left(\frac{s+1}{0.001s+1.8}\right)^2 |FRF_P\cdot Q_2^*|^2 \leq 1 \tag{5.43}$$

where $FRF_P$ is the frequency response of the uncertain plant ( i.e. the nominal plant model $G$ equipped with the multiplicative uncertainty weighting function $W_I$ )

The form of the identified model is incorporated in Equation 5.43. In practice, there is no need to parametrically identify the model because the problem can be solved solely in terms of the frequency response. This, therefore, constitutes a completely nonparametric method for the mixed sensitvity control problem.

The same settings for the LP algorithm are used in this uncertain system example with the order of power expansion as 60. By using the linear programming method, the optimal solution $Q^*$ is calculated as

$$Q_2^* = \frac{0.0003859s^5 + 0.1103s^4 + 8.253s^3 + 10.95s^2 + 10.59s + 3.292}{0.008644s^5 + 1.077s^4 + 8.23s^3 + 10.42s^2 + 9.379s + 2.889}$$

The resulting sensitivity functions are then

$$T_2^* = \frac{-0.006067s^6 - 1.692s^5 - 117.9s^4 + 714.3s^3 + 1010s^2 + 1086s + 353.6}{s^6 + 125.6s^5 + 1085s^4 + 2219s^3 + 2368s^2 + 1490s + 355.9}$$

$$S_2^* = \frac{1.006s^6 + 127.3s^5 + 1203s^4 + 1505s^3 + 1358s^2 + 403.3s + 2.304}{s^6 + 125.6s^5 + 1085s^4 + 2219s^3 + 2368s^2 + 1490s + 355.9}$$

and

$$V_2^* = \frac{0.04465s^5 + 12.76s^4 + 954.8s^3 + 1267s^2 + 1226s + 380.8}{s^5 + 124.5s^4 + 952.2s^3 + 1205s^2 + 1085s + 334.3}$$

It is also found that the internal stability is guaranteed due to the stability of the four functions $S_2^*$, $T_2^*$, $Q_2^*$ and $V_2^*$.

**iv) Compute the controller function $K$**

The controller is then calculated as

$$K_2 = \frac{0.04438s^6 + 12.73s^5 + 962.5s^4 + 2270s^3 + 2559s^2 + 1676s + 403}{s^6 + 126.5s^5 + 1195s^4 + 1495s^3 + 1350s^2 + 400.9s + 2.29}$$

In Figure 5.12, it can be seen that the primary sensitivity function $S^*$ is correctly bounded by the inverse of the weighting function $W_S$. The complementary sensitivity function $T^*$ is also correctly bounded by the inverse of the weighting function $W_T$ as shown in Figure 5.13. As shown in Figure 5.14, the step response

of the control system equipped with the controller $K_2$ has similar performance as it of the control system equipped with the controller $K_1$, which is computed by using the parametric plant model given as $G = \frac{1}{s+1}$. It is seen in the graph that the step response (blue dashed line) of the controller $K_1$ reaches the steady state in 4 seconds. At the same time at the fourth second, the step response of the controller $K_2$ also arrives its steady state. Furthermore, it is also observed in this figure that the two contorllers are robust to the uncertainty in the plant model (as seen in Figure 5.2).

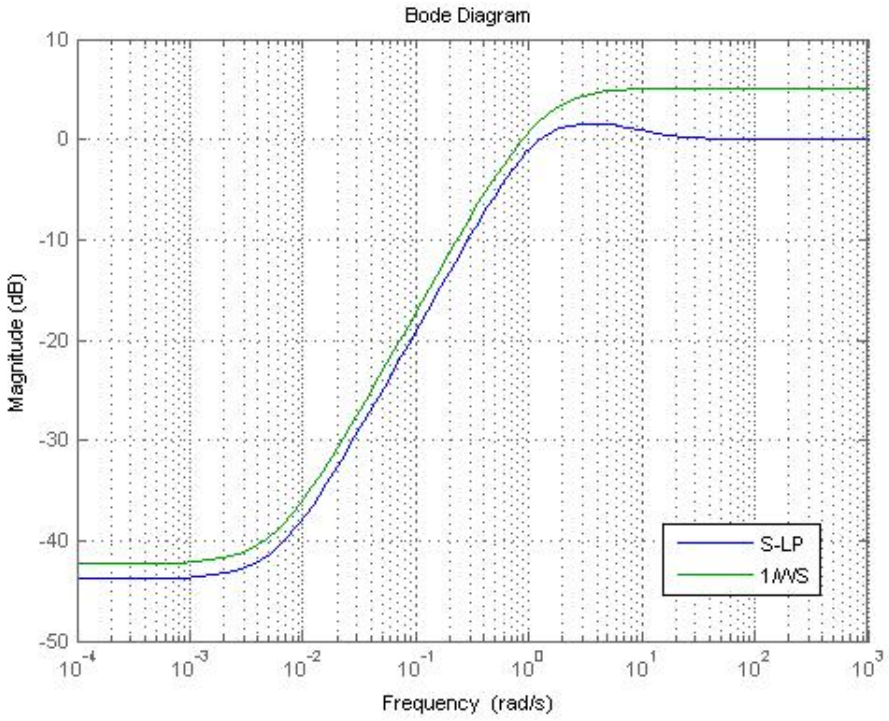



Figure 5.12: Comparison of Bode diagram of the primary sensitivity function $S_2$ and the weighting function $1/W_S$

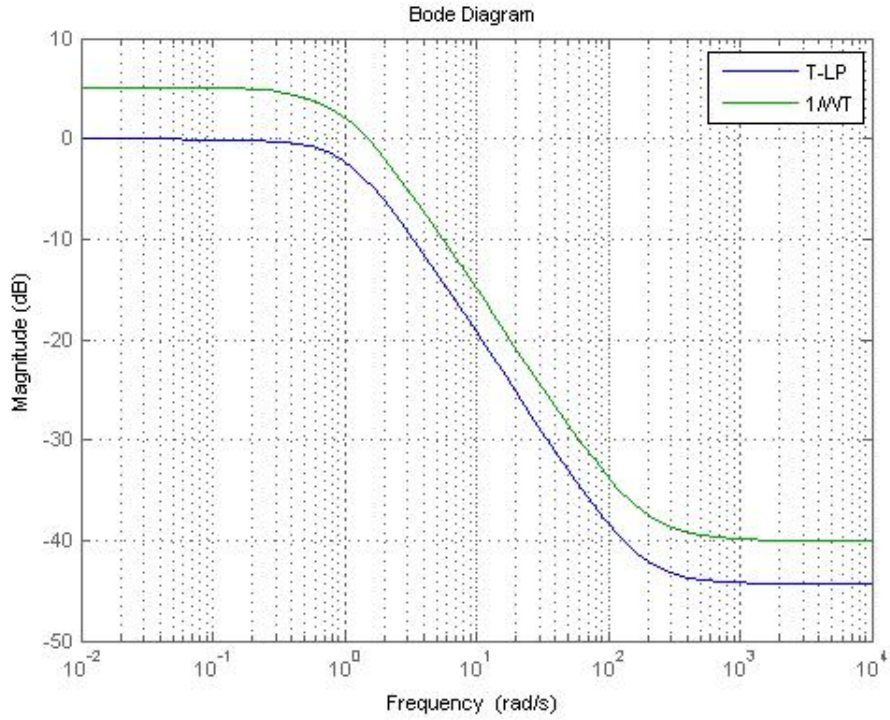



Figure 5.13: Comparison of the complementary sensitivity function $T_2$ and the inverse of the weighting function $W_T$

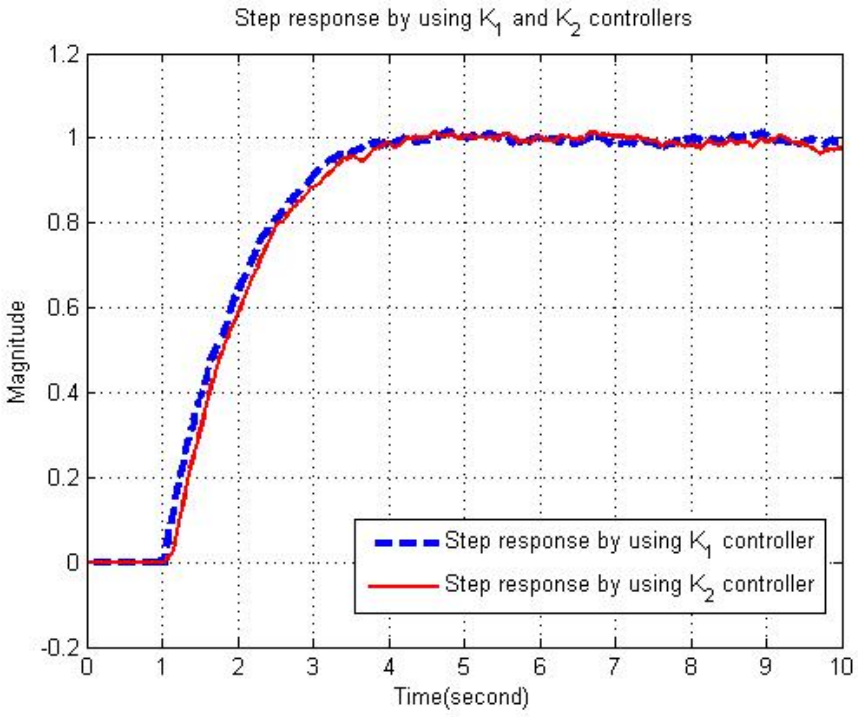



Figure 5.14: The system step responses with the controllers $K_1$ and $K_2$

In summary, the nonparametric method skips the process of the system identification. It is concluded that without knowing the transfer function of the plant (or any analytic form of the plant model), the proposed nonparametric $\mathbb{H}^\infty$ robust controller design method finds a controller which can be successfully implemented in the closed loop control system and has similar performance as the controller obtained by using the parametric method. The controller produced by the nonparametric method is also robust since it keeps the desired performance even if there is unmodelled dynamics in the plant or the external disturbances to the system.

## 5.5 Conclusions

- The optimization problem in the mixed sensitivity control problem is introduced. The mixed sensitivity problem is approximated as a root square problem in Equality 5.8. A nonparametric $\mathbb{H}^\infty$ controller design method in terms of Streit's algorithm [87] is proposed.

- The internal stability requirement in the control problem is discussed.

- The algorithm to solve the optimization problem in the mixed sensitivity control problem is summarized. Streit's algorithm [86] is implemented to tackle the mixed sensentivity control problem.

- The example taken from [83] is used to demonstrate the nonparametric method. We firstly found the resulting controller $K_1$ by using the given nominal plant $G$. The nonparametric method is then used to calculate the controller $K_2$.

- It is concluded that the nonparametric method finds the robust controller within the desired requirements. In terms of the frequency response, it does not need any system identification techinque. Furthermore, all the constraints are considered and the controller is found to be robust to the unmodelled dynamics or the external disturbances.

# Chapter 6

# An Application to the Engine Control Problem

## 6.1 Introduction

The nonparametric $\mathbb{H}^\infty$ control is formulated in Chapter 2 by means of the optimisation problem in Equation 2.1:

*Given a continuous positive-valued function* $\Gamma$ *in* $\mathbb{C}^N$ *where* $N$ *is the dimension of the problem, and a set of all the stable functions on the* $j\omega$ *axis ( denotes as* $\mathbb{RH}^\infty$ *, the space of all functions whose poles have negative parts. ), find the optimal function* $f^*(j\omega) \in \mathbb{RH}^\infty$ *such that*

$$\gamma = \inf_{f \in \mathbb{RH}^\infty} \sup_{\omega} \Gamma(\omega, f(j\omega)) \tag{6.1}$$

*where* $\omega$ *represents the frequency, and* $f(j\omega)$ *is the frequency response function.*

To solve the problem, the Disk Iteration (DI) method, Newton Iteration (NI) method, and the Linear Programming (LP) method are presented in detail in Chapter 2, 3, and 5 respectively. It is interesting to further investigate and compare the difference between these three algorithms (i.e. the DI, NI, LP methods) for nonparametric control. As presented in the previous chapters, the procedure to design the controller by using the $\mathbb{H}^\infty$ optimisation technique with different algorithms is used in this chapter.

Nevertheless, in this chapter, an example in [106] is used to analyze the DI, NI, and LP methods for nonparametric control. The control problem is the Peak-Pressure-Position (PPP) control problem in automotive engine control. Since the peak pressure position can be used to determine the optimal spark timing in order to generate the Maximum Brake Torque (MBT) [75, 76], the Engine Control

Unit (ECU) requires a controller to regulate the PPP by the Spark Advance (SA) demand. This is thus viewed as a command tracking problem. It is stated [17, 94] that ARX modelling is an effective linear identification structure for such problem. As a result, the transfer function of the plant identified by Ward [94] is used in this chapter :

$$G = \frac{0.6729s - 20.41}{s + 20.6}$$

It is noted that a RHP zero at $s = 30.3314$ exists in the plant's transfer function. To meet the internal stability requirement, in Helton's approach, the interpolation conditions such that $T(s) = A(s) + B(s)\,T_0(s)$ where $A(30.3314) = B(30.3314) = 0$ and $T_0 \in \mathbb{H}^\infty$ are required to meet. As a result, the interpolants $A$ and $B$ are chosen as $A = B = \frac{s-30.3314}{s+2}$. This is a parametric interpolation method used by Zhao [106]. However, a nonparametric (NP) identification technique without computing the pole and zero of the plant is possible to replace this parametric method and the frequency response is thus obtained by the NP identification method.

In this chapter, for comparison, the three proposed method, i.e. the DI method, the NI method, and the LP method, to solve an $\mathbb{H}^\infty$ control problem are analyzed. Beginning with the single sensitivity control, we discuss the resulting controller performances and the robustness of the system for this PPP control problem in Section 6.2. In Section 6.3, the three methods are compared again in terms of the mixed sensitivity control problem. The analysis of the methods are presented and summarized in Section 6.4.

## 6.2 Single Sensitivity Control

In [106], the weighting function for the single sensitivity problem is chosen as

$$W_S = \frac{s + 1.57}{1.57s + 0.01}$$

This weighting function will be used in this section to regenerate the results for analyzing the algorithms. 512 frequency points are chosen by using the mapping

$$\omega_i = b/\tan(\pi i/N)$$

where $i = 1, 2, 3, \cdots, N-1$ and the sample width $b = 1$.

The $\mathbb{H}^\infty$ single sensitivity control problem is thus formulated as

$$\gamma^* = \min_{s\in j\omega} \left|W_S\left(1-T^*\right)\right|^2 \min_{s\in j\omega} \left| \frac{s+1.57}{1.57s+0.01}\left(1-\frac{0.6729s-20.41}{s+20.6}Q^*\right)\right|^2 \tag{6.2}$$

The three methods to solve this optimization problem in 6.2 produce similar solutions as shown in Figure 6.1. The solutions of the DI, NI, and LP method are very close to each other. This implies that the problem is a convex optimization problem and all of them converge to the optimal solution.

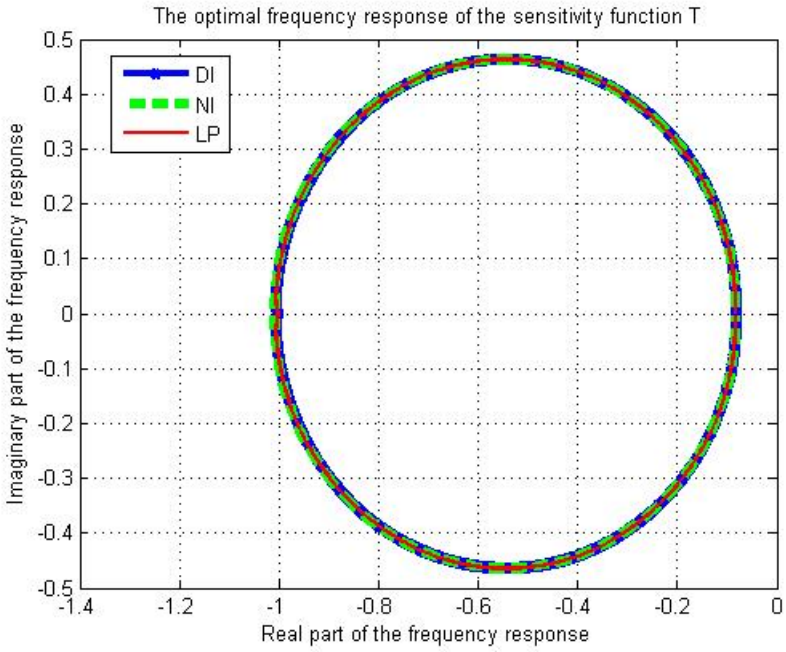



Figure 6.1: Comparison of the solutions by different analytic function methods

The resulting complementary sensitivity functions are then calculated as

$$T_{DI} = \frac{-0.0185s^5 - 0.09221s^4 + 13.86s^3 + 180.8s^2 + 9.349s + 0.1697}{0.3346s^5 + 13.02s^4 + 135.2s^3 + 187.4s^2 + 9.493s + 0.1707}$$

$$T_{NI} = \frac{-0.006624s^5 - 0.2541s^4 + 7.346s^3 + 195.9s^2 + 1.33s + 0.03086}{0.119s^5 + 8.727s^4 + 138.8s^3 + 196.6s^2 + 1.35s + 0.0308}$$

$$T_{LP} = \frac{-0.02995s^5 + 0.04204s^4 + 20.45s^3 + 176.7s^2 + 7.047s + 0.165}{0.5429s^5 + 17.49s^4 + 138.7s^3 + 181.6s^2 + 7.179s + 0.1659}$$

and the primary sensitivity functions are

$$S_{DI} = \frac{0.3531s^5 + 13.11s^4 + 121.4s^3 + 6.664s^2 + 0.1444s + 0.0009898}{0.3346s^5 + 13.02s^4 + 135.2s^3 + 187.4s^2 + 9.493s + 0.1707}$$

$$S_{NI} = \frac{0.1256s^5 + 8.981s^4 + 131.5s^3 + 0.7264s^2 + 0.0194s - 0.00005823}{0.119s^5 + 8.727s^4 + 138.8s^3 + 196.6s^2 + 1.35s + 0.0308}$$

$$S_{LP} = \frac{0.5728s^5 + 17.45s^4 + 118.3s^3 + 4.906s^2 + 0.1317s + 0.000907}{0.5429s^5 + 17.49s^4 + 138.7s^3 + 181.6s^2 + 7.179s + 0.1659}$$

Figure 6.2a shows the Bode diagrams of these complementary sensitivity functions and Figure 6.2b illustrates the Bode diagram of these primary sensitivity functions. It is observed that all of the sensitivity functions $S_{DI}$, $S_{NI}$, and $S_{LP}$ are bounded by the inverse of the weighting function $W_S$.

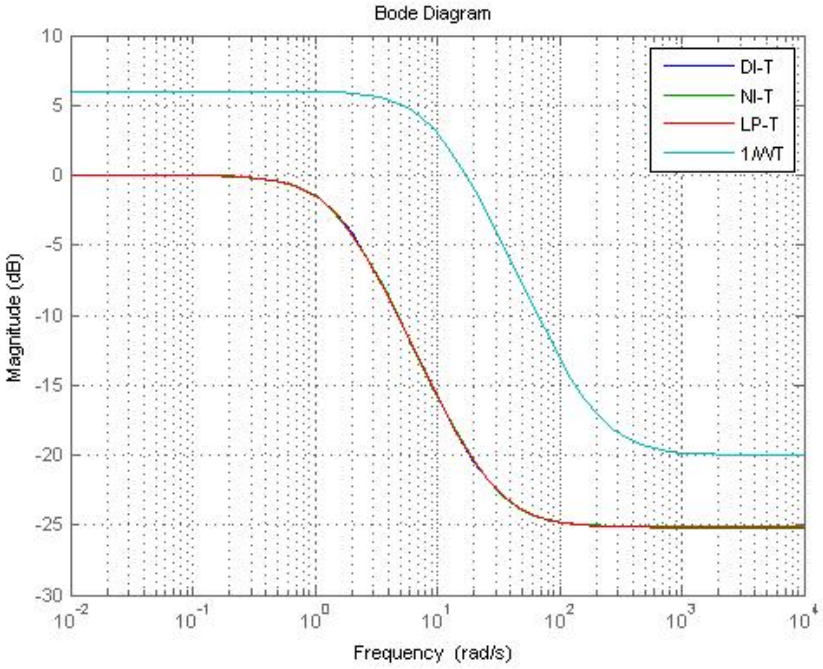
The controllers are then computed as

$$C_{DI} = \frac{-0.05045s^4 - 3.426s^3 - 77.37s^2 - 581.9s - 14.2}{s^4 + 49.59s^3 + 598.9s^2 + 18.74s + 0.1758}$$

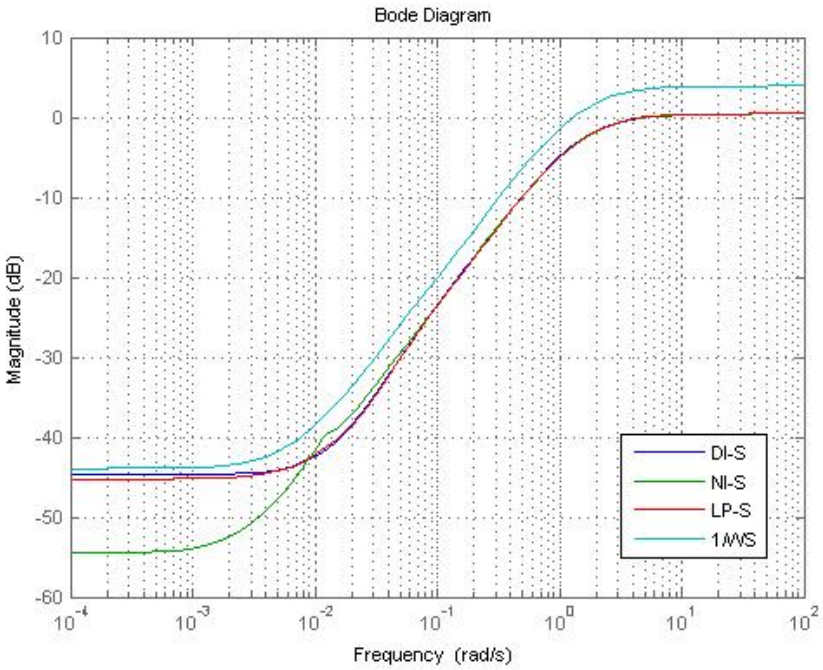
$$C_{NI} = \frac{-0.04944s^4 - 3.074s^3 - 64.42s^2 - 455s - 10.93}{s^4 + 43.69s^3 + 477.9s^2 + 23.6s + 0.3265}$$

$$C_{LP} = \frac{-0.05049s^4 - 3.504s^3 - 80.48s^2 - 613.2s - 15.51}{s^4 + 51.15s^3 + 631s^2 + 20.11s + 0.1921}$$

and the simulation of the corresponding step responses in Matlab/Simulink [65] are shown in Figure 6.3. It is seen that the settling time of these responses are all around 6 seconds and there are no overshoots in the transient response.




(a) Bode diagram of the complementary sensitivity functions $T_{DI}$, $T_{NI}$, $T_{LP}$ in the single sensitivity control problem




(b) Bode diagram of the primary sensitivity functions $S_{DI}$, $S_{NI}$, $S_{LP}$ in the single sensitivity control problem

Figure 6.2: Comparison of the sensitivity function $S$ and $T$ in their Bode diagrams

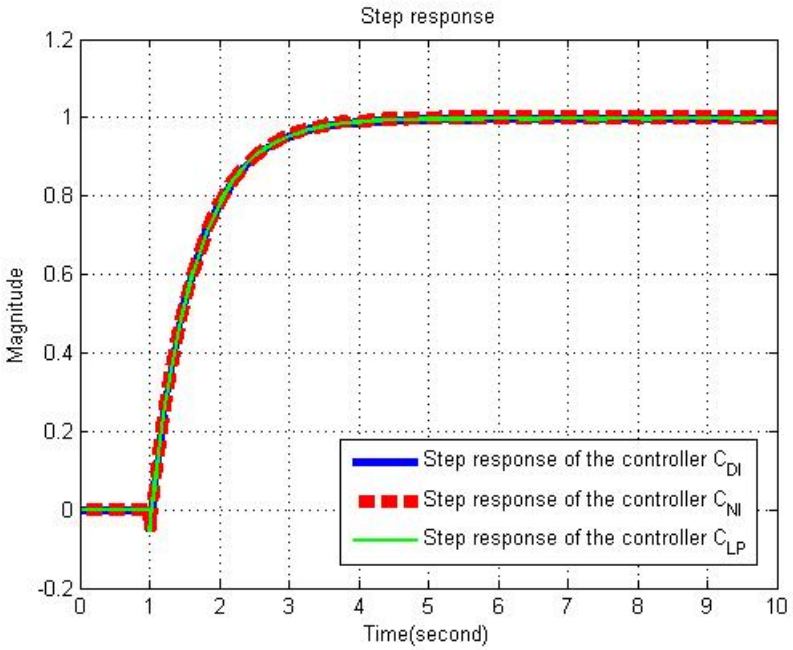



Figure 6.3: Step responses of the controllers $C_{DI}$, $C_{NI}$ ,and $C_{LP}$

## 6.3 Mixed Sensitivity Control

The mixed sensitivity control problem for the PPP control problem is formulated as [106]

$$\gamma^* = \inf_{Q^* \in \mathbb{H}^\infty} \sup_{\omega} \left| W_S \left(1 - GQ^*\right) \right|^2 + \left| W_T \left(GQ^*\right) \right|^2$$

where $W_S$ and $W_T$ are the two weighting functions chosen as

$$W_S = \frac{s + 1.57}{1.57s + 0.01}$$

$$W_T = \frac{0.2s + 1}{0.01s + 2}$$

The same frequency points in the previous section are selected to be equally spaced on the unit circle. That is, 1024 frequency points are chosen by the equation

$$\omega_i = b / \tan\left(\pi i / N\right)$$

where $i = 1, 2, 3, \cdots, N-1$ and $b = 1$.

In Figure 6.4, it is seen that the three methods, i.e. the DI, NI and LP methods, produce the similar solutions which are very close to each other. This again implies the convexity of this control problem and the local optimum is the

same as the global optimum no matter the analytic function method we use.

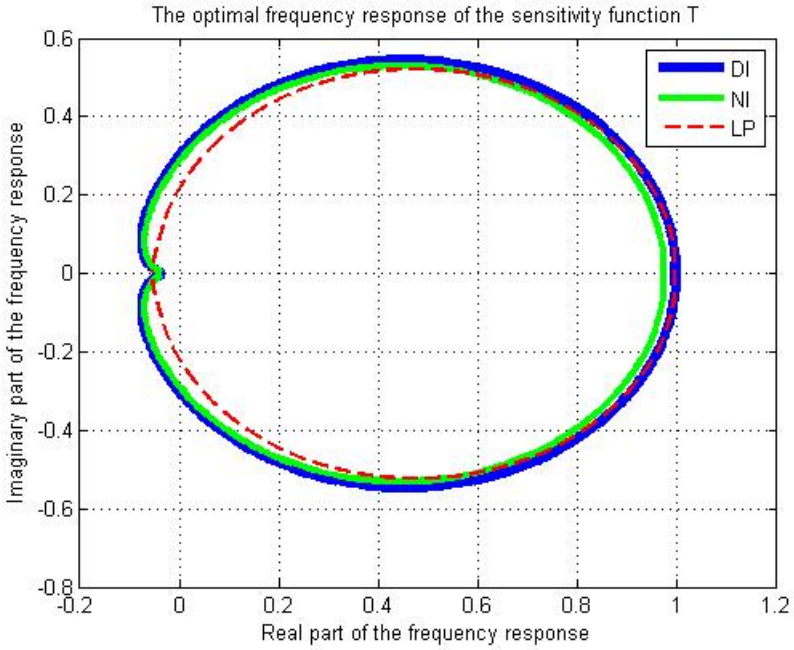



Figure 6.4: Comparison of the frequency respons of the analytic solutions $Q^*$ by different methods

The complementary sensitivity functions and the primary sensitivity functions are then calculated as

$$T_{DI} = \frac{-0.0003611s^5 - 0.02064s^4 + 0.1945s^3 + 16.6s^2 + 199.1s + 5.549}{0.01109s^5 + 0.6385s^4 + 13.95s^3 + 123.1s^2 + 202.4s + 5.572}$$

$$T_{NI} = \frac{-0.0005278s^5 - 0.02384s^4 + 0.3749s^3 + 19s^2 + 191s + 5.535}{0.0146s^5 + 0.7821s^4 + 15.5s^3 + 124.7s^2 + 198.9s + 5.682}$$

$$T_{LP} = \frac{-0.01982s^5 - 0.06931s^4 + 14.53s^3 + 175.8s^2 + 13.53s + 0.3128}{0.3569s^5 + 13.36s^4 + 132.6s^3 + 185.2s^2 + 13.78s + 0.3145}$$

$$S_{DI} = \frac{0.01145s^5 + 0.6592s^4 + 13.75s^3 + 106.5s^2 + 3.286s + 0.02362}{0.01109s^5 + 0.6385s^4 + 13.95s^3 + 123.1s^2 + 202.4s + 5.572}$$

$$S_{NI} = \frac{0.01513s^5 + 0.806s^4 + 15.13s^3 + 105.8s^2 + 7.921s + 0.147}{0.0146s^5 + 0.7821s^4 + 15.5s^3 + 124.7s^2 + 198.9s + 5.682}$$

$$S_{LP} = \frac{0.3767s^5 + 13.43s^4 + 118.1s^3 + 9.417s^2 + 0.2485s + 0.001636}{0.3569s^5 + 13.36s^4 + 132.6s^3 + 185.2s^2 + 13.78s + 0.3145}$$

Figure 6.5a shows the three resulting complementary sensitivity functions and Figure 6.5b illustrates the three primary sensitivity functions. It is found in the figures that the complementary sensitivity functions $T_{DI}$, $T_{NI}$, and $T_{LP}$ are bounded by the weighting function $1/W_T$. The primary sensitivity functions $S_{DI}$, $S_{NI}$, and $S_{LP}$ are also bounded by the inverse weighting function $1/W_S$.

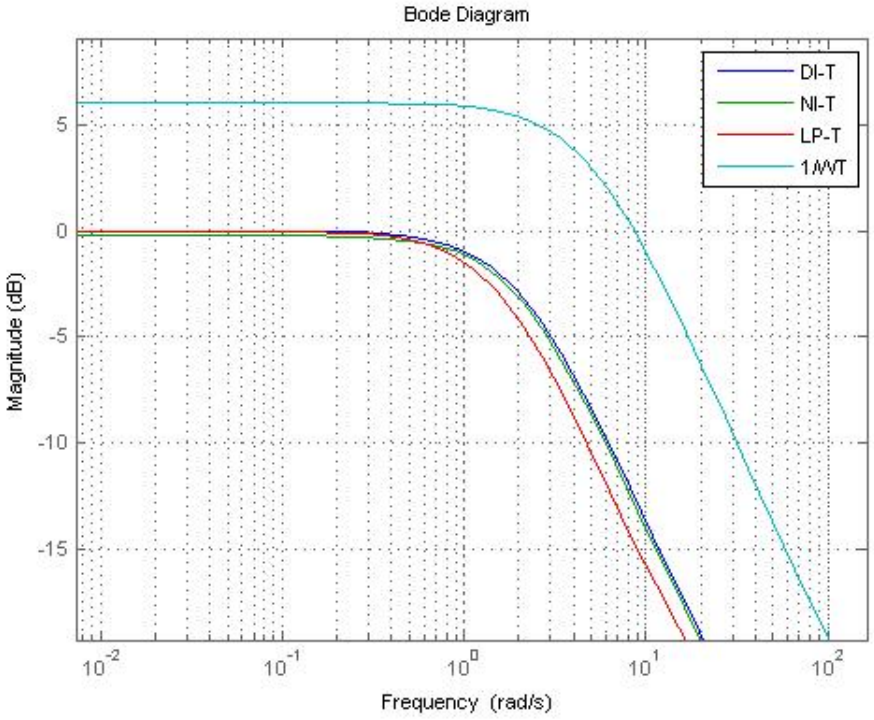
Thus, the controllers are computed as

$$C_{DI} = \frac{-0.04687s^5 - 5.066s^4 - 183.6s^3 - 2895s^2 - 1.759e04s - 490.4}{s^5 + 57.57s^4 + 1201s^3 + 9303s^2 + 287s + 2.063}$$

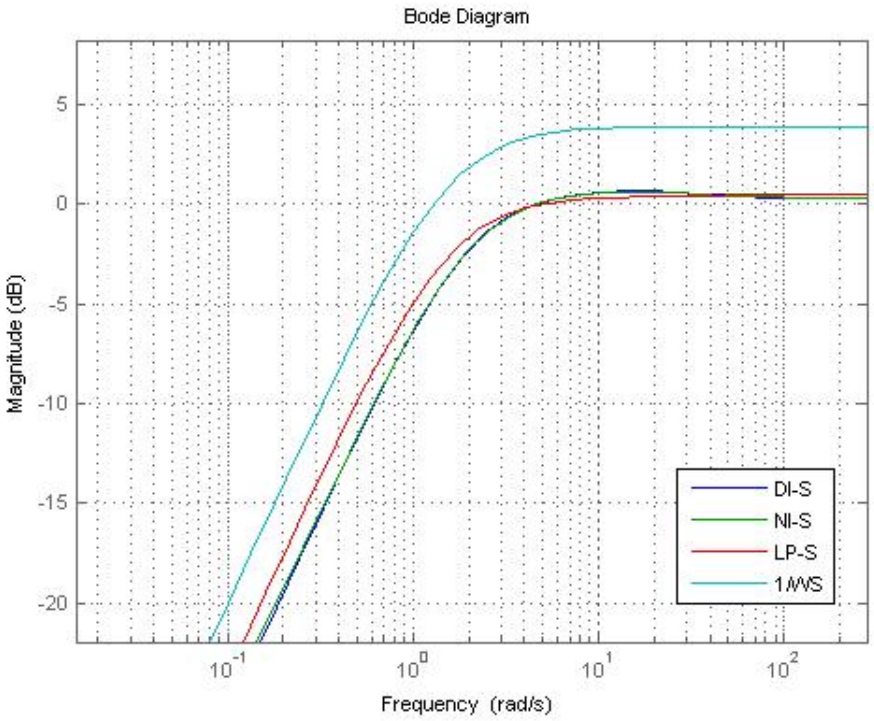
$$C_{NI} = \frac{-0.05185s^5 - 4.983s^4 - 162.6s^3 - 2307s^2 - 1.278e04s - 369.7}{s^5 + 53.28s^4 + 1000s^3 + 6991s^2 + 523.6s + 9.718}$$

$$C_{LP} = \frac{-0.07818s^5 - 4.256s^4 - 77.42s^3 - 474s^2 - 36.34s - 0.8409}{s^5 + 35.64s^4 + 313.5s^3 + 25s^2 + 0.6595s + 0.004343}$$

Figure 6.6 shows the simulated step responses of the controllers $C_{DI}$, $C_{NI}$, and $C_{LP}$. It is observed from the plot that the step responses of the controller $C_{DI}$ ,$C_{NI}$ and $C_{LP}$ are close to each other. All the three step responses reach the steady state at around 4 seconds. Compared to the step responses in the single sensitivity control problem in Figure 6.3, it is observed that the solution to the mixed sensitivity problem may have better performance in the time domain. Moreoever, since there is an additional constraint, e.g. $W_T$, in the mixed sensitivity problem, more performance requirements can be included in the consideration during the optimization process. The mixed sensitivity control is potential to deal with more general engineering control problems than the single sensitivity control.




(a) Bode diagrams of the complementary sensitivity functions $T_{DI}$, $T_{NI}$, $T_{LP}$




(b) Bode diagrams of the complementary sensitivity functions $S_{DI}$, $S_{NI}$, $S_{LP}$

Figure 6.5: Comparison of the sensitivity functions $S$ and $T$ in their Bode diagrams

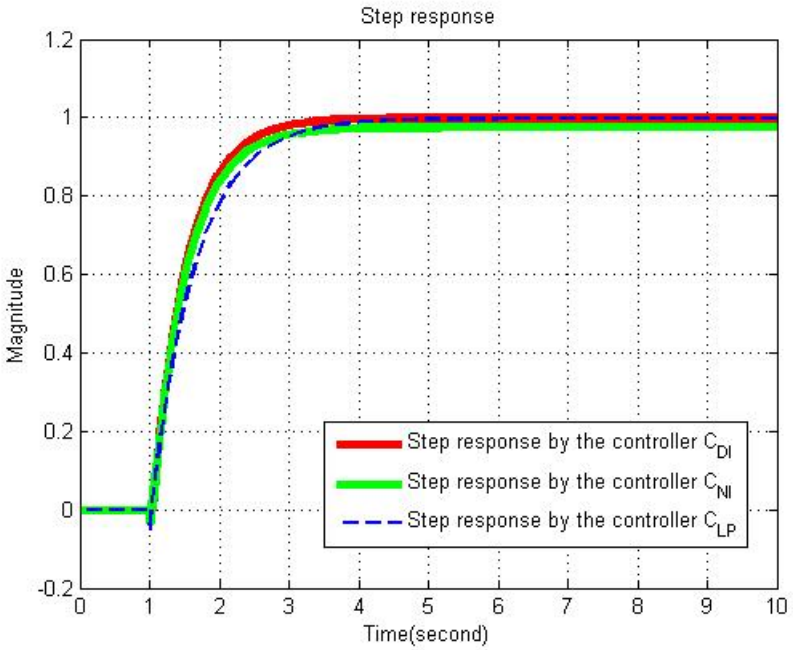



Figure 6.6: Step responses of the controllers $C_{DI}$, $C_{NI}$ ,and $C_{LP}$

## 6.4 Discussions

### 6.4.1 Selection of the initial guess for the NI method

It is pointed out in the previous chapters that NI method is very sensitive to the initial points. The same conclusion can also be observed in this example. In the single sensitivity control problem, the NI method converges to the same solution as the others from the initial points at $(0, 0)$. However, in the mixed sensitivity control problem, the NI method only produces a similar solution compared with the others when starting from around the points close to the optimal points. Other points away from this point, e.g. larger than the distance of 0.02, will result in divergent solutions by the NI method. This represents a significant drawback if using the current implementation of the NI method. The reason for its sensitivity to the initial points is possibly from the inaccuracy of inverting the joint operator in the Matlab implementation. Furthermore, if the optimization problem is not convex, i.e. the local optimum is not neccessarily the global optimum, because the three optimization methods, the NI, DI, and LP methods, are algorithms to search for the local optimum, and so the global optimal solution may be lost during the optimization process. When using these analytic function methods, it is essential for the designer to examine the properties of the calculated solution before accepting the resulting controller.

### 6.4.2 Pros and cons of the LP method

The LP method is known to be superior in the sense of its capability to deal with the contraints both in the frequency domain and the time domain [40]. It is also mentioned in the results of [12] that the LP method requires more computational effort than the other two iteration methods. However, it is still useful to adopt the LP method because the LP method is in principle more accurate than the other optimization methods. Moreover, as mentioned in [43], the LP method is still valid when there exists any pole of the plant near the unit circle while the DI and NI methods may fail in such cases. The disadvantage of using the LP method is from the internal storage of the matrices. Normally, as in the LP method the functions are expanded to high order series in the form of matrices, this data preprocessing requires a huge amount of memory and results in higher computing power. Sometimes the computation needs more than 8GB memory and several hours of running on a personal computer with a dual core processor to accomplish the job when the expansion order is up to 60 and the discretization $p = 1024$.

Fortunately, the obstable of such long computing time for the LP method is being reduced by the improvements in computer technology. Furthermore, in Matlab, the codes for the interior point method has been optimized for solving linear programming problems to achieve better accuracy and less computing time in large sparse problems [32]. It is found in the examples that the computing time are not significantly long. In the experience of running the Matlab programs, the elapsed time is mostly within 30 minutes with 512 sampling points.

In addition, although it is reported in [86] to require the discretization number $p$ larger than 1024, it has been observed experimentally in the work for this thesis that the solution to the linear programming problem only needs $p = 32$ for the satisfactory accuracy $10^{-4}$ with the analytic solution value $\gamma_{analytic} = 0.6673955751046930$. Comparison of different discretization numbers $p$ for the mixed sensitivity control problem in this chapter is shown in Table 6.4.1 by using the expansion order $n = 10$. If a sufficiently large enough number of sampling points are used, the discretization number does not require to be the recommended number of 1024. For the optimization process in this automotive PPP control, $p = 32$ is fairly enough for the required accuracy $10^{-4}$.

| The discretization number $p$ | The optimal value $\gamma^*$ | Computing time (seconds) | Accuracy ($\left\|\frac{\gamma^* - \gamma_{analytic}}{\gamma_{analytic}}\right\|$) |
|---|---|---|---|
| 4 | 0.6673056475992780 | 5.6068 | 0.000134744 |
| 8 | 0.6673056475920020 | 15.4236 | 0.000134744 |
| 16 | 0.667305647619287 | 40.0480 | 0.000134744 |
| 32 | 0.667359618216096 | 125.8325 | 0.0000538764 |
| 64 | 0.667371939973236 | 330.7124 | 0.000035414 |
| 128 | 0.667384748308564 | 764.9893 | 0.0000162225 |
| 256 | 0.667388190078782 | 4343.7941 | 0.0000110654 |

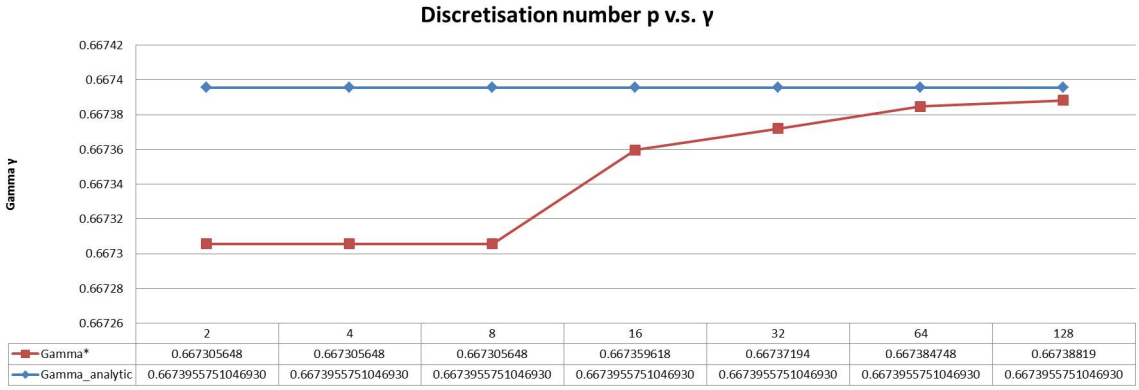



Table 6.4.1: Comparison of the different discretization numbers $p$ for the S.I.S.O. mixed sentivity control problem

The other influencing factor in the LP method is the order of power expansion of the function $Q$, i.e. the number $n$ in Equation 5.24. It is expected that the higher the order $n$ that is used, the more accurate the solution will be. Nevertheless, the computing time and the acccuracy are to be traded off to achieve an acceptable time frame. For a fixed value of $p = 16$, the comparison of the different orders of the expansion series is shown in Table 6.4.2. As a result, it is suggested to set the order $n$ as 60 and $p = 32$ for similar mixed sensitivity control problems with the accuracy of $10^{-5}$.

| | Expansion order $n$ | Optimal Value $\gamma^*$ | Computing Time (seconds) | Accuracy ($\left\|\frac{\gamma^*-\gamma_{analytic}}{\gamma_{analytic}}\right\|$) |
|---|---|---|---|---|
| $p=16$ | 3 | 0.672414512919204 | 15.2204 | 0.0003887022464567 |
| | 5 | 0.670549727441141 | 23.0906 | 0.0006234178891308 |
| | 10 | 0.667305647619287 | 37.6418 | 0.0001347439041560 |
| | 20 | 0.666158106139847 | 64.8095 | 0.0000160914639161 |
| | 30 | 0.666043375297477 | 108.4114 | 0.0000158840520315 |
| | 60 | 0.665869361604564 | 340.7438 | 0.0000011011161933 |
| | 80 | 0.665839261753931 | 431.9747 | 0.0000000628141906 |
| | 100 | 0.665833678127228 | 704.6933 | 0.0000000057071264 |
| | Expansion order $n$ | Optimal Value $\gamma^*$ | Computing Time (seconds) | Accuracy ($\left\|\frac{\gamma^*-\gamma_{analytic}}{\gamma_{analytic}}\right\|$) |
| $p=32$ | 3 | 0.672527986430957 | 61.4150 | 0.0002200125436547 |
| | 5 | 0.670913079447928 | 73.4415 | 0.0000818838734037 |
| | 10 | 0.667359618216096 | 117.0368 | 0.0000538764264228 |
| | 20 | 0.666158106152579 | 180.2518 | 0.0000160914448039 |
| | 30 | 0.666043375087611 | 244.5649 | 0.0000158843671201 |
| | 60 | 0.665869361924706 | 783.1476 | 0.0000011006354057 |
| | 80 | 0.665839261630026 | 944.8358 | 0.0000000630002790 |
| | 100 | 0.665833677248884 | 1771.2715 | 0.0000000070262908 |

Table 6.4.2: Comparison of different orders $n$ for the SISO mixed sensitivity control problem

### 6.4.3 Comparison of the methods

It can be observed from the above results that the solutions by the three analytic function methods are close to each other. Therefore, a choice may come down to the comparison on the computing power required for each method.
Table 6.4.3 shows the computing time for the single sensitivity problem and the mixed sensitivity problem by each analytic function method. It is observed that the NI method needs the least computing power but it requires a careful choice of the initial guess. The DI method is effective but increased number of iterations makes the computation longer in time. It is found that the LP method, as expected, requires the longest computing time. However, the computation in the linear programming algorithm may be improved by using third-party optimization softwares, such as TomLab[52] or CVX [33, 34]. These softwares are employed to speed up the process of optimization and may also improve the accuracy of the solution. For example, by using CVX in the single sensitivity problem, the computing time can be significantly reduced from 381.5326 seconds to 2.3550 seconds, which makes use of the method more convenient.

In addition, the interpolation method proposed by Helton [47] requires the computation of the unstable zero and pole of the plant. In the nonparametric approach, introducing the sensitivity function $Q$ to guarantee the internal stability

| Optimization Algorithm | Single Sensitivity Problem | Mixed Sensitivity Problem |
|---|---|---|
| | Computing time (second) | Computing time (second) |
| NI | 105.9434 | 94.9621 |
| DI | 302.5748 | 314.3073 |
| LP | 381.5326 | 656.1859 |
| LP by CVX | 2.3550 | 7.3107 |

Table 6.4.3: Comparison of the computing times by DI, NI, and LP methods

does not need any parametric interpolation, which leads to a fully nonparametric controller design method - a major outcome of this thesis.

## 6.5 Conclusions

- A control problem for powertrain engine PPP control problems is presented. The application of the $\mathbb{H}^\infty$ controller design method to the Peak-Pressure-Position (PPP) control problem is addressed in this chapter. It is shown in the results that, in the single sensitivity weighting method, the optimal sensitivity function is correctly bounded by the inverse of the weighting function and the resulting three optimal closed-loop step responses have a good settling time around of 6 seconds. Moreover, in the mixed sensitivity weighting method, the results demonstrate that all the sensitivity functions are bounded by the inverses of the respective weighting functions. Compared to the closed-loop step responses in the single sensitivity problem, the step responses in the mixed sensitivity problem has an improved settling time of 4 seconds.

- The accuracy of the solution is found to be relatively insensitive to the discretization number $p$. Although it is suggested in [86] to use $p = 1024$ to deal with most optimization problems, the error between the numerical and analytic solutions is acceptable in the problems studied if $p$ is selected as 32. The order of expansion $n$, however, is more influential than the number $p$. Considering the computing cost and the usage of memory storage, $n = 60$ is suggested to be good enough for similar mixed sensitivity control problems.

- The three optimization algorithms are compared and analyzed. It is found that the solutions by the analytic function methods are close to each other. The DI method needs inner iterations to approximate the objective function to the closest circular form. This rises the cost in computation for the DI method. Although the NI method theoretically has a second convergence rate, the convergence of the solutions is very sensitive to the initial points.

However, this may be due to the Matlab implementation method such as the method of inversion used. The LP method may require a higher amount of memory to store the variable matrices, which also results in longer computing time on a restricted memory computer. However, some third-party optimization software for Matab can significantly reduce the time of computation in the LP method.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

Several algorithms are investigated in the thesis for developing nonparametric controller design methods. This chapter draws the conclusions and findings on the implementations in the previous chapters, and suggests the directions for future research work according to the content of the thesis. Matlab implementations of various analytic function optimization methods primarily due to Helton, Merino, and Walker[38, 39, 46, 48] are implemented for the first time. The use of these methods in application to real world and automotive control problems are investigated. The primary conclusion of this thesis is that a fully nonparametric method for stable multivariable systems is proposed and demonstrated based on analytic function optimization methods and this is successfully applied to the mixed sensitivity control problem. The technique is also applied to an automotive PPP control problem. In summary, the nonparametric control approach has the advantage of skipping the approximating process of parametric system identification and only relies on obtaining the Fourier coefficients of the system impulse response function from the plant input-output data. In the DI and NI method and the frequency response itself in the LP method, the approach potentially saves time and cost in the controller design stage. In addition, it is sometimes neccessary to obtain highly accurate and reliable linear models for some complex engineering systems in terms of their frequency responses. This is most directly obtained via the Fourier coefficients from a Blackman-Tukey analysis [13] of the plant input-output data. Thus, the proposed nonparametric control approach offers a path to design a robust controller directly using the data involved in obtaining the frequency response from the measured data.

The specific conclusions from each chapter are

- In Chapter 1, fundamental $\mathbb{H}^\infty$ feedback control theory is introduced. Gen-

erally, the $\mathbb{H}^\infty$ control method is a controller synthesis with guaranteed performance and stability requirements. Such methods transform the control problem to a mathematical optimization problem and find the solution to the optimization problem by using special-purposed optimization algorithms. The overview on the development of the $\mathbb{H}^\infty$ method is presented in Section 1.1. The introduction to major automotive engine control problems is outlined in Section 1.2. The overview to each chapter of this thesis is given in Section 1.3. The contributions of the work in this thesis are then presented in Section 1.4.

- In Chapter 2, the Helton-Merino Disk Iteration (DI) method is presented. In Section 2.1, the optimization problem in the $\mathbb{H}^\infty$ control scheme is reviewed. The DI method is based on the solution to the corresponding Nehari problem. The spectral factorization technique is required in the DI method. To make this approach nonparametric, the existing technique of parametric spectral factorization must be replaced by nonparametric methods. Two nonparametric spectral factorization methods are reviewed in the beginning of Section 2.2. The solution to the Nehari problem is provided by using Nehari-commutant-lifting formula as discussed in Section 2.2.2. However, the solution to the Nehari problem only exists if the objective function is in the circular form. Therefore, it is essential to find the closest circular form of the objective function. The inner iteration procedure to approximate to the circular form is implemented in terms of a quadratic spline interpolation method as proposed by Helton and Merino. This Matlab implementation of the DI method is described in details in Section 2.2.

- In Chapter 3, the Helton-Merino-Walker Newton Iteration (NI) method is presented and analysed. The optimality conditions for the $\mathbb{H}^\infty$ optimization problem are stated in Section 3.1. These conditions are the foundation of the derivation of the NI method. The NI method is based on the solution to the optimality conditions of the $\mathbb{H}^\infty$ optimization problem by using the Toeplitz-plus-Hankel operator. The theoretical study of the NI method and the algorithm are given in Section 3.2. Unlike the DI method, the NI method does not require any approximation to a specific form of objective function. Furthermore, the NI method has a second order convergence rate in theory. These imply that the NI method is potentially a better approach than the DI method to solving the $\mathbb{H}^\infty$ optimization problem. To investigate this approach, a Matlab implementation of the NI method is presented at the end of Section 3.2.

- In Chapter 4, the above two algorithms are compared and analyzed with three numerical examples. In Section 4.1, 4.2 and 4.3, the formulations of the examples are presented. It is seen that if the objective function contains only one variable, both methods (i.e. the DI and NI methods) have similar performance in terms of the computing time and the number of iterations used. However, if the objective function has more than one variable, it is possible for the NI method to converge faster than the DI method due to the NI method's second order convergence rate. This finding is addressed in Section 4.4. Unfortunately, it is also found experimentally that with the Matlab implementation, the NI method is very sensitive to the initial guess. If the initial points are not close to the optimum, the algorithm may not be convergent. Some of the observations in the examples are shown in Section 4.5.

- In Chapter 5, a linear programming nonparametric approach to the design a robust controller in $\mathbb{H}^\infty$ mixed sensitivity control scheme using Hadamard $\mathbb{H}^\infty$-Frobenius norm [92, 91] is presented. This approach has the significant advantage over conventional maximum singular value approaches to $\mathbb{H}^\infty$ design in that the Hadamard weights allows each element of the closed-loop transfer function to be weighted individually and thus gives the designers direct control over each element allowing, for example, exact decoupling. In Section 5.2, the mixed sensitivity control problem is reformulated in terms of the $\mathbb{H}^\infty$ norm as a linear optimization problem. To meet the internal stability conditions of the controlled system, the interpolation proposed by Helton [48] and the method to replace the complementary sensitivity function $T$ by the sensitivity function $Q$ are outlined in the beginning of the same section. The nonparametric $\mathbb{H}^\infty$ controller design method using the replacement method for stable systems is also proposed in the same section. The algorithm requires that the designer first selects the upper bounds of the performance functions $S$ and $T$ as the corresponding weighting functions $W_S$ and $W_T$. By substituting the frequency responses of the functions $W_S$, $W_T$, and the measured freequency response data from the plant $G$, the designer can fully formulate the discrete version of the optimization problem. By using the appropriate optimization algorithm, the frequency responses of the optimal sensitivity function $Q^*$ can be found. The optimal $\mathbb{H}^\infty$ controller is then computed by $K = Q\left[I - GQ\right]^{-1}$ and then filtered in a rational function format for digital controller implementation.
  In Section 5.3, the optimization (LP) method based on Streit's algorithm [86] originally investigated by Helton and Sideris [40] is described. The

LP method may be considered a better optimization algorithm for the $\mathbb{H}^\infty$-Frobenius norm optimization problem because of its superior accuracy and capability of integrating the time-domain and frequency-domain performance requirements. However, the LP method significantly requires more computing time and memory storage compared to the other two methods. Nevertheless, its problem of computing power may be reduced by using third-party optimization softwares, e.g. TomLab [52] or CVX[33, 34]. An example to demonstrate the LP method is shown in Section 5.4.

- In Chapter 6, the three optimization methods are applied to an automotive control problem for comparison. In Section 6.1, this automotive control problem is viewed as a command tracking problem since the Peak-Pressure-Position (PPP) of the engine crank is correlated to the Spark Advance (SA). To optimize the timing of combustion, the transfer function of the plant and the weighting functions are used from [106]. The results in Section 6.2 show that, as required, in the single sensitivity weighting problem, the optimal sensitivity functions by the three proposed methods, i.e. the DI, NI, and LP methods, are bounded by the inverse of the weighting function $W_S$ and the three optimal step responses have settling times around 6 seconds. In the mixed sensitivity weighting problem in Section 6.3, it is found that, all the sensitivity functions are bounded as required by the inverses of the corresponding weighting functions $W_S$ and $W_T$. Compared to the step responses in the single sensitivity problem, the step responses in this mixed sensitivity control have smaller settling time with a value around 4 seconds. In investigating the three methods, the accuracy of the solutions is found to be unrelated to the discretization number $p$. Although it is suggested in [86] that $p = 1024$ is generally required for general optimization problems, it is observed from the experiments that within the acceptable accuracy $p$ need only be selected as 32. The order of expansion $n$, however, is more influential than the number $p$. Considering the computing power and the memory usage, $n = 60$ is suggested to be enough for similar optimization problems. The three optimization algorithms are thus compared and analyzed in Section 6.4. The analytic solutions from the three methods are shown to be similar to each other. It is understood that the DI method needs multi-level iterations for approximating the objective function to the closest circular form. Although the NI method has a second-order convergence rate in theory, it is very sensitive to the initial points. The LP method, on the other hand, requires larger amounts of memory to store the matrices, which results in the longest computing time when used on a

limited memory PC.

## 7.2 Perspectives of Future Work

- The nonparametric approach to the control problem has several advantages for engineers in the design of robust controllers. More applications of this approach may be investigated in the future to see if they can benefit from this approach including both SISO and MIMO control systems. Comparison with the existing parametric methods would then be a point for future research.

- The extension of the nonparametric approach to MIMO problems by using the suggested nonparametric Youla-Kucera Q-parameterization method for internal stability can be further developed. The applications of the method may be employed for more general control problems in engineering.

- For the internal stability requirement, the nonparametric Youla-Kucera Q-parameterization method used in this thesis only works on stable plants. To cope with the unstable plants for control, the Youla-Kucera Q-parameterization method [60, 102] can be implemented by considering the left coprime factorization of the plant $G$ : $G = M^{-1}N$ where $M$ and $N$ are the coprime factors [93]. The parameterization of all stabilizing controllers in a feedback control loop is then given as $K = [Y - QN]^{-1}[X + QM]$ where $X$ and $Y$ are stable matrices which satisfy the Bezout identity $NX + MY = I$ and $Q$ is any stable function. In principle, the NI and LP method should allow solution of the $G = M^{-1}N$ factorization and the stable $X$ and $Y$ in the Bezout identity equation. A nonparametric coprime factorization method would significantly enhance the applicability of nonparametric control.

- The selection of appropriate weighting functions may be studied for further investigation for powertrain control. We know that the weighting functions represent the required performance requirements. It is also seen in the thesis that the choice of weighting functions determines the availability of an analytic solution. It is, therefore, important to establish the relationships between the weighting functions and the performance requirements in the frequency or time domain.

- It is possible to design two-degree-of-freedom controllers in the proposed nonparametric approach. The two-degree-of-freedom control scheme as shown in Figure 7.1 is known to be the best control strategy for achieving

significantly improved tracking performance. In other words, the tracking errors in 2-degree-of-freedom control systems are generally significantly less than those of 1-degree-of-freedom systems. Therefore, the formulation of the equivalent $\mathbb{H}^{\infty}$ optimization problem for the 2-degree-of-freedom control could be further investigated and it may be possible for the NI and LP optimization algorithms to be used to support the application.

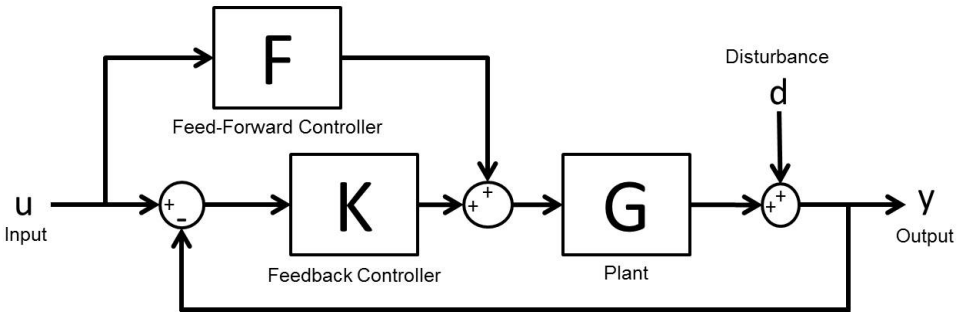



Figure 7.1: Configuration of a 2-degree-of-freedom control sytsm

# Appendix A. Proof of Nonvanishing Gradient

This appendix is to prove the assumption of the optimality conditions in Section 3.1, Chapter 3. The assumption excludes the trivial situation so that the strict local directional optimizer can be well defined. The theorem is proved in [84].

**Theorem 9.** *If the gradient ( all the partial derivatives ) of a function $f$ defined in domain $\mathbb{R}^N$ vanishes, then the function $f$ is a constant in the whole domain.*

*Proof.* Suppose $\nabla f = 0$ in the domain $D$ of $\mathbb{R}^N$, i.e. for $N = 2$, $f_x^{'}(x,y) = f_y^{'}(x,y) = 0$, $\forall\,(x,y) \in D$ and there are two points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ such that $f(x_1, y_1) \neq f(x_2, y_2)$. Since the derivatives of $f$ are continuous and are zero in $D$, the distance between $P_1$ and $P_2$ with an interior point $(a, b)$ on the line $\overline{P_1P_2}$ can be written by Mean-Value theorem as

$$f(x_1, y_1) - f(x_2.y_2) = f_x^{'}(a,b)(x_1 - x_2) + f_y^{'}(a,b)(y_1 - y_2)$$

The right side of the equation equals zero because $f_x^{'}(x,y) = f_y^{'}(x,y) = 0$ and contradicts the left side according to the assumption $f(x_1, y_1) \neq f(x_2, y_2)$. This proves the theorem. □

# Appendix B. Derivation of Equation 3.22 in the Newton Iteration Method

In this appendix, we derive the Equation 3.22 as proposed in [39]. The operator equation, i.e. Equation 3.21, is

$$T'_{f,\beta}\begin{pmatrix}\varepsilon\\ \delta\end{pmatrix}=\begin{pmatrix}P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi\beta\right)\right)\left(A\bar{\varepsilon}+B\varepsilon\right)+2a\Re\left(\chi\delta\right)\right]\\ P_{\mathbb{H}^{2\perp}_{1}}\left[2\Re a^{T}\varepsilon\right]\end{pmatrix}$$

where $a=\frac{\partial}{\partial z}\Gamma\left(\cdot,f\right)$, $A=\frac{\partial^2}{\partial z\partial\bar{z}}\Gamma\left(\cdot,f\right)$, $B=\frac{\partial^2}{\partial z^2}\Gamma\left(\cdot,f\right)$ and $A$ is assumed to be real positive ( i.e. $\bar{A}=A$ ).

Note that in the following derivation, the complex rules will be used :
$\Re\left(z\right)=\frac{1}{2}\left(z+\bar{z}\right)$,
$\frac{1}{\chi}=\bar{\chi}$ for $\chi\triangleq e^{j\theta}$,
$\bar{\bar{\chi}}=\chi$
as well as the transformations stated in [39] P.852 :

$$P_{\mathbb{H}^{2\perp}_{N,0}}\left[\bar{M}\bar{\eta}\right]=CP_{\mathbb{H}^{2}_{N}}\left[M\eta\right]$$

where $C$ represents the complex conjugate operator $C:f\rightarrow\bar{f}$, and

$$P_{\mathbb{H}^{2\perp}_{N,0}}\left[\chi M\eta\right]=\chi P_{\mathbb{H}^{2\perp}_{N}}\left[M\eta\right]\Rightarrow\bar{\chi}P_{\mathbb{H}^{2\perp}_{N,0}}\left[\chi M\eta\right]=P_{\mathbb{H}^{2\perp}_{N}}\left[M\eta\right]$$

Introducing $\omega\triangleq1+2\Re\left(\chi\beta\right)$ where $\omega$ is assumed to be positive ( i.e. $\bar{\omega}=\omega$ ), we have

$$\begin{aligned}T'_{f,\beta}&=\begin{pmatrix}P_{\mathbb{H}^{2\perp}_{N,0}}\left[\left(1+2\Re\left(\chi\beta\right)\right)\left(A\bar{\varepsilon}+B\varepsilon\right)+2a\Re\left(\chi\delta\right)\right]\\ P_{\mathbb{H}^{2\perp}_{1}}\left[2\Re a^{T}\varepsilon\right]\end{pmatrix}\\&=\begin{pmatrix}P_{\mathbb{H}^{2\perp}_{N,0}}\left[\omega\left(A\bar{\varepsilon}+B\varepsilon\right)+2a\Re\left(\chi\delta\right)\right]\\ P_{\mathbb{H}^{2\perp}_{1}}\left[2\Re a^{T}\varepsilon\right]\end{pmatrix}\end{aligned}$$

The above equation can be further expressed in the following :

$$
\begin{aligned}
T'_{f,\beta} &= \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\omega A\bar{\varepsilon}+\omega B\varepsilon+2a\frac{\overline{\chi\delta}+\chi\delta}{2}\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[2\frac{\overline{a^T\varepsilon}+a^T\varepsilon}{2}\right] \end{pmatrix} \\
&= \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\omega A\bar{\varepsilon}+\omega B\varepsilon+a\bar{\chi}\bar{\delta}+a\chi\delta\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[\bar{a}^T\bar{\varepsilon}+a^T\varepsilon\right] \end{pmatrix} \\
&= \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\omega A\bar{\varepsilon}+a\bar{\chi}\bar{\delta}\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[\bar{a}^T\bar{\varepsilon}\right] \end{pmatrix} + \begin{pmatrix} P_{\mathbb{H}^{2\perp}_{N,0}}\left[\omega B\varepsilon+a\chi\delta\right] \\ P_{\mathbb{H}^{2\perp}_{1}}\left[a^T\varepsilon\right] \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&= \begin{pmatrix} CP_{\mathbb{H}^2_N}\left[\left(\bar{\omega}\bar{A}\right)\varepsilon\right]+CP_{\mathbb{H}^2_N}\left[\left(\chi\bar{a}\right)\delta\right] \\ \bar{\chi}P_{\mathbb{H}^{2\perp}_{1,0}}\left[\chi\bar{a}^T\bar{\varepsilon}\right]=\bar{\chi}CP_{\mathbb{H}^2_1}\left[\bar{\chi}a^T\varepsilon\right] \end{pmatrix} + \begin{pmatrix} \chi P_{\mathbb{H}^{2\perp}_N}\left[\bar{\chi}\omega B\varepsilon\right]+\chi P_{\mathbb{H}^{2\perp}_N}\left[a\delta\right] \\ P_{\mathbb{H}^{2\perp}_1}\left[a^T\varepsilon\right] \end{pmatrix} \\
&= CP_{\mathbb{H}^2_N}\left[\begin{pmatrix} \bar{\omega}\bar{A} & \chi\bar{a} \\ \bar{\chi}\bar{\chi}a^T & 0 \end{pmatrix}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix}\right]+P_{\mathbb{H}^{2\perp}_N}\left[\begin{pmatrix} \omega B & \chi a \\ a^T & 0 \end{pmatrix}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix}\right] \\
&= CP_{\mathbb{H}^2_N}\left[\begin{pmatrix} I_N & 0 \\ 0 & \bar{\chi} \end{pmatrix}\begin{pmatrix} \omega A & \chi\bar{a} \\ \bar{\chi}a^T & 0 \end{pmatrix}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix}\right] \\
&\quad +P_{\mathbb{H}^{2\perp}_N}\left[\begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \bar{\chi}\omega B & a \\ a^T & 0 \end{pmatrix}\begin{pmatrix} \varepsilon \\ \delta \end{pmatrix}\right] \\
&= \begin{pmatrix} I_N & 0 \\ 0 & \bar{\chi} \end{pmatrix} CT_{\begin{pmatrix} \omega A & \chi\bar{a} \\ \bar{\chi}a^T & 0 \end{pmatrix}}\begin{bmatrix} \varepsilon \\ \delta \end{bmatrix}+\begin{pmatrix} \chi I_N & 0 \\ 0 & 1 \end{pmatrix}\mathcal{H}_{\begin{pmatrix} \bar{\chi}\omega B & a \\ a^T & 0 \end{pmatrix}}\begin{bmatrix} \varepsilon \\ \delta \end{bmatrix}
\end{aligned}
$$

This is Equation 3.22 in Section 3.2.

# Appendix C. Quadratic Spline Function

To derive the quadratic spline interpolation function, for three points are on the spline function $f(x)$ and that we have the interpolation conditions :

$$f(x_1) = y_1,\ f(x_2) = y_2,\ f(x_3) = y_3$$

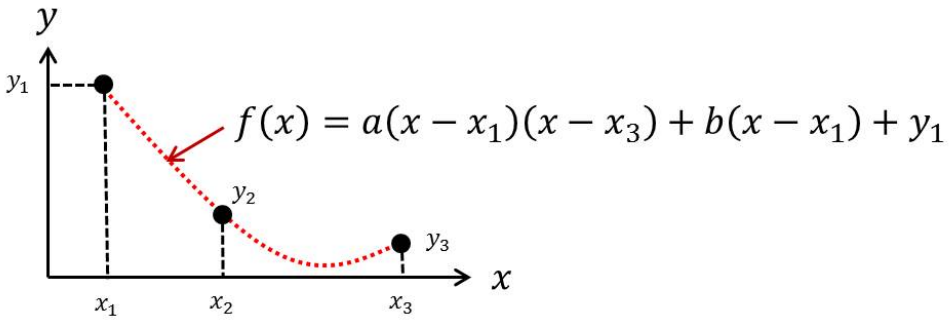



Figure. 301: Quadratic spline function

By carefully choosing the representation of $f(x)$ shown in Figure. 301 such that

$$f(x) = a(x - x_1)(x - x_3) + b(x - x_1) + y_1$$

and noting that

$$y_1 = f(x_1)$$

we require that

$$\begin{aligned} y_2 = f(x_2) &= a(x_2 - x_1)(x_2 - x_3) + b(x_2 - x_1) + y_1 \\ y_3 = f(x_3) &= b(x_3 - x_1) + y_1 \end{aligned}$$

The neccessary value of the variable $b$ can be immediately obtained :

$$b = \frac{y_3 - y_1}{x_3 - x_1}$$

and then the neccessary value of variable $a$ can be obtained as

$$\begin{aligned} a &= \frac{1}{(x_2 - x_1)(x_2 - x_3)} \left[ y_2 - y_1 - b(x_2 - x_1) \right] \\ &= \frac{(y_2 - y_1)(x_3 - x_1) - (y_3 - y_1)(x_3 - x_1)}{(x_2 - x_1)(x_2 - x_3)(x_3 - x_1)} \end{aligned}$$

# Appendix D. Golden Section Search Method

The golden section search method proposed by Kiefer [58] is the technique of finding the local extremum in the unimodal maximization problem. That is, it is the algorithm to solve the maximization problem : $\max f(x)$ subject to $a \leq x \leq b$ where only one maximum exists and $f(x)$ is a smooth function.

The method is easily revised in order to solve the minimization problem. The idea is to compute the function values of the interior points with the values at the bounds, to reduce the interval until the extreme point remains in the sufficiently small range.

First of all, suppose we are given a function $f(x)$ in an interval $[a, b]$ and its extreme point (minimum) appears at a point $x^* \in [a, b]$, i.e. $f(x^*) = \min_{[a,b]} f(x)$, $f(x)$ is a *unimodal function* if $f(x)$ is strictly descreasing from $a$ to $x^*$ and strictly increasing from $x^*$ to $b$ ( or reverse the monotonicity on each side of $x^*$ if the extreme point is a maximum.)

Next, suppose we are given a unimodal function $f(x)$ defined in the interval $[a, b]$ as shown in Figure. 401. We can pick two points, say $x_1$ and $x_2$, in the interval and evaluate their function values as $f_1$ and $f_2$.

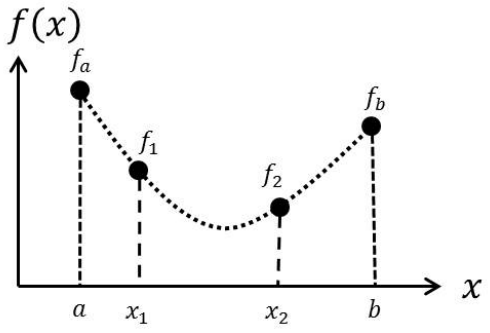



Figure. 401: A unimodal function $f(x)$ in the interval $[a, b]$

With regard to the four function values $f_a$, $f_1$, $f_2$ and $f_b$, as shown in Figure. 402, it is known that for the cases :

1. $f_1 > f_2$
   If $f_1 > f_2$, the optimum should be either on the right to $f_2$ (Figure. 402 (a)) or between $f_1$ and $f_2$ (Figure. 402 (b)). In either case, it is understood that the optimum must be between the interval $[x_1, b]$. In other words, the lower bound is shifted from $a$ to $x_1$.

2. $f_1 < f_2$
   Another possible sitation is that if $f_1 < f_2$, the optimum should be either on the left to $f_1$ (Figure. 402 (c)) or between $f_1$ and $f_2$ (Figure. 402 (d)). Again in either case, it is known that the optimum must be between the interval $[a, x_2]$. The upper bound is shifted from $b$ to $x_2$.

3. $f_1 = f_2$
   If $f_1 = f_2$, we can immediately observe that the optimum must be in the interval $[f_1, f_2]$. This situation which results in a big improvement in reducint the interval is rare. Therefore, such case is arbitrarily considered as being the first case $f_1 > f_2$ and the interval is updated from $[a, b]$ to $[x_1, b]$.

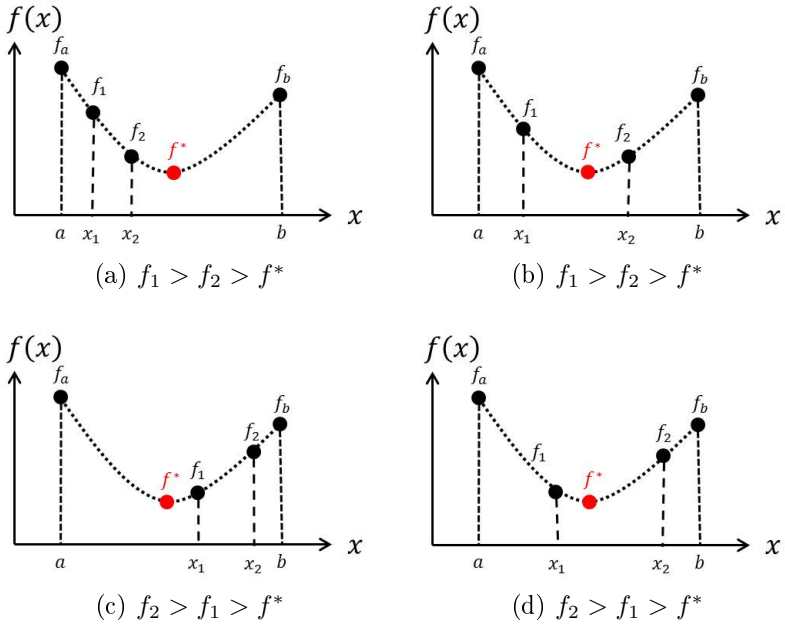



Figure. 402: $f_1$ and $f_2$ in the interval $[a, b]$

The important issue in the GSS method is the selection of the interior points. In Figure. 403, $a$ and $b$ are the left bound and right bound of current interval, and $c$ and $x$ are an interior point in $[a, b]$ such that $\delta = \frac{b-c}{b-a}$ and a new point in the segment $[c, b]$ for the next iteration. We then have the relations :

$$1 - \delta = \frac{c - a}{b - a}$$

$$\delta = \frac{b-c}{b-a}$$

$$z = \frac{x-c}{b-a}$$

As a result, the next feasible interval may be $[a, x]$ or $[c, b]$ depending on the function value $f(x)$. To minimize the worst case possibility ( i.e. $z < 0$ or $z > \delta$ ), we choose to make the size of $[a, x]$ equal the size of $[c, b]$, i.e.

$$1 - \delta + z = \delta \tag{401}$$

Note that the strategy of choosing the interior point is the same in the previous iteration. Therefore, the fraction $1 - \delta$ should be the same as the fraction $\frac{z}{\delta}$, i.e.

$$1 - \delta = \frac{z}{\delta} \tag{402}$$

Substituting $z$ in Equation 402 by Equation 401, we have

$$1 - \delta = \frac{2\delta - 1}{\delta}$$

i.e.

$$\delta^2 + \delta - 1 = 0 \tag{403}$$

We can immediately have the solution to Equation 403 as $\delta = \frac{-1+\sqrt{5}}{2} \approx 0.61803$. The value of $\delta$ is the negative value of the conjugate solution to the characteristic equation of the well known *Golden Ratio* ($\frac{1+\sqrt{5}}{2}$). In other words, we now choose the optimized distances of $(a, x_2)$ and $(x_1, b)$ as $\frac{-1+\sqrt{5}}{2}(a-b)$. Therefore, we have $x_1 = b - \delta(b-a)$ and $x_2 = a + \delta(b-a)$, i.e. $x_1 = a + (1-\delta)(b-a)$ and $x_2 = a + \delta \cdot (b-a)$.

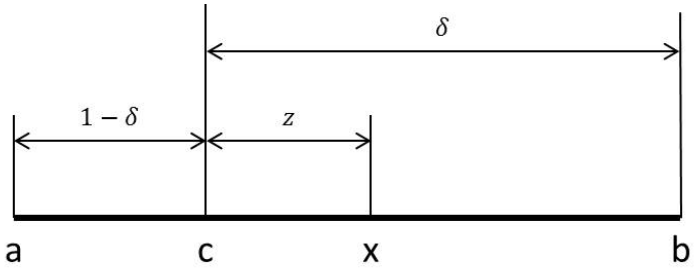



Figure. 403: Selection of the interior point

In summary, the algorithm for finding the minimum of a unimodal function in the interval $[a, b]$ is as follows :

**Algorithm.D. .1** Golden Section Search

1. Define the interval $[a, b]$
2. Select two interior points $x_1$ and $x_2$ by $x_1 = a + (1 - \delta)(b - a)$ and $x_2 = a + \delta(b - a)$ where $\delta = \frac{-1+\sqrt{5}}{2}$
3. Evaluate $f_1 = f(x_1)$ and $f_2 = f(x_2)$
4. If $f_1 \geq f_2$, reset the interval as $[x_1, b]$. Otherwise, reset the interval as $[a, x_2]$.
5. If the length of the interval is smaller than the tolerance, i.e. $|a - b| < \varepsilon$, stop the program. Otherwise, repeat the computation from Step 2.

# References


[1] A. Abass. *Automotive Driveline Control by Nonparametric QFT Methods.* PhD thesis, University of Liverpool, 2011.

[2] M. Abu-Khalaf, J. Huang, and F.L. Lewis. *Nonlinear H2/H-Infinity Constrained Feedback Control : A Practical Design Approach using Neural Networks.* Springer, 2006.

[3] V.M. Adamjan, D.Z. Arov, and M.G. Krein. Infinite Hankel and Generalized Caratheodory-Fejer and M. Riesz Problems. *Functional Analysis Application*, 2:1–18, 1968.

[4] V.M. Adamjan, D.Z. Arov, and M.G. Krein. Analytic Properties of Schmidt Pairs for Hankel Operator and the Generalized Schur-Takagi Problem. *Mathematics of USSR-Sb*, 15:15–78, 1972.

[5] V.M. Adamjan, D.Z. Arov, and M.G. Krein. Infinite Block Hankel Matrices and Their Connection with the Interpolation Problem. *American Mathematical Society Translations*, 111(2):133–156, 1978.

[6] I. Akira and K. Hitoshi. *Linear Time Varying Systems and Sampled-Data Systems.* Springer, 2001.

[7] K.J. Astrom and R.M. Murray. *Feedback Systems : An Introduction for Scientists and Engineers.* Princeton University Press, 2008.

[8] G. Balas, A. Packard, R. Chiang, and M. Safonov. *MATLAB Robust Control Toolbox 3 User's Guide.* The MathWorks, Inc., 2010.

[9] J.A. Ball and J.W. Helton. A Beurling-Lax Theorem for the Lie Group U(m,n) which Contains Most Classical Interpolation Theory. *Journal of Operator Theory*, 9:107–142, 1983.

[10] F.L. Bauer. Ein direktes iterations verfahren zur hurwitz-zerlegung eines polynorms. *Archiv Elektrische Uebertragung*, 9:285–290, 1955.

[11] F.L. Bauer. Beitrage zur entwichlung numerischer verfahren fur programmgesteuerte rechenanlagen ii - direkte faktorisierung eines polynoms. *Sitzung Berticht Bayerischen Akademische Wissenschaften*, pages 163–203, 1956.

[12] J. Bence, J.W. Helton, and D.E. Marshall. Optimization over H-infinity. In *IEEE Conference on Decision Control*, 1986.

[13] R. B. Blackman and J.W. Tukey. *The Measurement of Power Spectra from the Point of View of Communications Engineering.* New York , Dover, 1958.

[14] S. Boyd and C. Baratt. *Linear Controller Design : Limits of Performance.* Prentice-Hall, 1991.

[15] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[16] C. Caratheodory and L. Fejer. Uber den zusammenhang der extremen von harmonischen funktionen mit ihren koeffizienten und tiber den picard-landauschen satz. *Rend. Circ. Mat. Palermo*, 32:218–239, 1911.

[17] S.D. Carroll. *Control of S.I. Engines using In-Cylinder Pressure.* PhD thesis, University of Liverpool, 2003.

[18] Gu D-W, P.H. Petkov, and M. M. Konstantinov. *Robust Control Design with MATLAB.* Springer, 2005.

[19] J. Davis and R.G. Dickinson. Spectral Factorization by Optimal Gain Iteration. *SIAM Jounral of Applied Mathematics*, 43:289–301, 1983.

[20] J.C. Doyle. Structured Uncertainty in Control System Design. In *IEEE Conference on Decision and Control*, volume 24, pages 260–265, 1985.

[21] J.C. Doyle, K. Glover, P. P. Khargonekar, and B.A. Francis. State-Space Solutions to Standard H2 and H-infinity Control Problems. *IEEE Transactioins on Automatic Control*, AC-34(8):831–847, 1989.

[22] Ecotrons. Small engine management system. `http://www.ecotrons.com/products/small_engine_management_system/`.

[23] C. Foias, H. Ozbay, and A. Tannenbaum. *Robust Control of Infinite Dimensional Systems : Frequency Domain Methods.* Springer, 1996.

[24] B.A. Francis. *A Course in H-infinity Control Theory*. Springer-Verlag, 1987.

[25] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson, 2008.

[26] G. Galdos, A. Karimi, and R. Longchamp. H-infinity Controller Design for Spectral MIMO Models by Convex Optimization. *Journal of Process Control*, 20(10):1175–1182, December 2010.

[27] D. Garcia, A. Karimi, and R. Longchamp. Data-Driven Controller Tuning using Frequency Domain Specifications. *Industrial & Engineering Chemistry Research*, 45(12):4032–4042, 2006.

[28] J.B. Garnett. *Bounded Analytic Functions*. Springer, 2007.

[29] G.Gu, P. Khargonrkar, and B. Lee. Approximation of Infinite-Dimensional Systems. *IEEE Transactioins on Automatic Control*, 34(6):610–618, 1989.

[30] K. Glover and D. McFarlane. Robust Stabilization of Normalized Coprime Factor Plant Descriptions with H-infinity Bounded Uncertainty. *IEEE Transactioins on Automatic Control*, AC-34(8):821–830, 1989.

[31] I.C. Gohberg and M.G. Krein. Systems of Integral Equations on a Half Line with Kernels Depending on the Difference of Arguments. *American Mathematical Society Translations*, 14:217–288, 1960.

[32] J. Gondzio and T. Terlaky. A Computational View of Interior-Point Methods for Linear Programming. Technical report, Polish Academy of Science and Eotvos University, 1994.

[33] Michael Grant and Stephen Boyd. Graph Implementations for Nonsmooth Convex Programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[34] Michael Grant and Stephen Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.0 beta. `http://cvxr.com/cvx`, September 2013.

[35] L. Guzzella and C.H. Onder. *Introduction to Model and Control of Internal Combustion Engine Systems*. Springer, 2004.

[36] T.J. Harris and J.H. Davis. An Iterative Method for Matrix Spectral Factorization. *SIAM Jounral on Scientific and Statistical Computing*, 13(2):531–540, March 1992.

[37] J. W. Helton. Optimal Frequency Domain Design vs. an Area of Several Complex Variables. In *Robust Control of Linear Systems and Nonlinear Control - Proceedings of the International Symposium MTNS-89, Volume II*, volume 2, pages 33–59, 1990.

[38] J. W. Helton, O. Merino, and T.E. Walker. Conditions for Optimality over H-infinity : Numerical Algorithms. In *Decision and Control*, 1992.

[39] J. W. Helton, O. Merino, and T.E. Walker. Algorithms for Optimizing over Analytic Functions. *Indiana University Mathematics Journal*, 42(3):839–874, 1993.

[40] J. W. Helton and A. Sideris. Frequency Response Algorithm for H-infinity Optimization with Time Domain Constraints. *IEEE Transactions on Automatic Control*, 34(4):427–434, 1989.

[41] J. William Helton and Orlando Merino. Anopt and Optdesign. Internet: http://www.math.ucsd.edu/ anopt/, 1998.

[42] J.W. Helton. Worst Case Analysis in the Frequency Domain : The H-infinity Approach to Control. *IEEE Transactions on Automatic Control*, AC-30(12):1154–1170, 1985.

[43] J.W. Helton. Optimization Over Spaces of Analytic Functions and the Corona Problem. *Jounral of Operator Theory*, 15:359–375, 1986.

[44] J.W. Helton and M.R. James. *Extending H-infinity Control to Nonlinear Systems*. SIAM, 1999.

[45] J.W. Helton and D. Marshall. Frequency Domain Design and Analytic Selections. *Indiana University Mathematics Journal*, 39(1):157–184, 1990.

[46] J.W. Helton and O. Merino. A Novel Approach to Accelerating Newton's Method for Sup-Norm Optimization Arising in H-infinity Control. *Journal of Optimization Theory*, 78(3):553–578, 1993.

[47] J.W. Helton and O. Merino. *Classical Control Using H-infinity Methods*. SIAM, 1998.

[48] J.W. Helton, O. Merino, and T.J. Walker. Optimization Over Analytic Functions whose Fourier Coefficients are Constrained. *Integral Equations and Operator Theory*, 22(4):420–439, Dec 1995.

[49] J.W. Helton and H.J. Woerdeman. Symmetric Hankel Operators : Minimal Norm Extensions and Eigenstructures. *Linear Algebra and its Applications*, 185:1–19, 1993.

[50] P. Henrici. *Applied and Computational Complex Analysis.* Wiley, New York, 1974.

[51] J. Heywood. *Internal Combustion Engine Fundamentals.* McGraw-Hill, 1988.

[52] Kenneth Holmstrom. The Tomlab Optimization Environment in Matlab, 1999.

[53] I. Horowitz. *Synthesis of Feedback Systems.* New York Academic Press, 1963.

[54] D. Hrovat and J. Sun. Models and Control Methodologies for IC Engine Idle Speed Control Design. *Control Engineering Practice*, 5(8):1093–1100, 1997.

[55] R. Jurgen. *Automotive Electronic Handbook.* McGraw-Hill, 1995.

[56] A. Karimi and G. Galdos. Fixed-order H-infinity Controller Design for Non-parametric Models by Convex Optimization. *Automatica*, 46:1388–1394, 2010.

[57] A. Karimi, M. Kunze, and R. Longchamp. Robust Controller Design by Linear Programming with Application to a Double-Axis Positioning System. *Control Engineering Practice*, 15:197–208, 2007.

[58] J. Kiefer. Sequential Minimax Search for a Maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

[59] U. Kiencke and L. Nielsen. *Automotive Control Systems.* Springer, 2005.

[60] V. Kucera. *Discrete Linear Control: The Polynomial Equation Approach.* Wiley, Chichester, UK, 1979.

[61] V. Kucera. Factorization of Rational Spectral Matrices : A Survey of Methods. In *Iternational Conference on Control*, volume 2, pages 1074–1078, March 1991.

[62] B.C. Kuo and F. Golnaraghi. *Automatic Control Systems*. Wiley, New York, 2002.

[63] MATLAB. *Control System Toolbox*. MathWorks, 2012.

[64] MATLAB. *Optimization Toolbox User's Guide*. MathWorks, 2012.

[65] MATLAB. *version 7.14.0 (R2012a)*. The MathWorks Inc., 2012.

[66] D. McFarlane and K. Glover. *Robust Controller Design using Normalized Coprime Factor Plant Descriptions*, volume 138. Springer-Verlag, 1990.

[67] D. McFarlane and K. Glover. A Loop Shaping Design Procedure using H-infinity Synthesis. *IEEE Transactions on Automatic Control*, 37(6):759–769, June 1992.

[68] O. Merino. *Optimization Over Spaces of Analytic Functions*. PhD thesis, University of California at San Diego, 1988.

[69] B. Sz Nagy, C. Foias, and L. Kerchy. *Harmonic Analysis of Operators on Hilbert Space*. Springer, 2 edition, 2010.

[70] Z. Nehari. On Bounded Bilinear Forms. *Annals of Mathematics*, 65(1):153–162, 1957.

[71] R. Nevanlinna. Uber beschrankte funktionen, die in gegeben punkten vorge schriebene werte annehmen. *Annales Academia Scientiarum Fennica*, 13:27–43, 1919.

[72] G. Opfer. Solving Complex Approximation Problems by Semiinfinite - Finite Optimization Techniques : A Study on Convergence. *Numerische Mathematik*, 39:411–420, 1982.

[73] G. Pick. Uber die beschrankungen analytischer funktionen, welche durch vorgegebene funktionswerte bewirkt sind. *Mathematische Annalen*, 77:7–23, 1916.

[74] S.J. Poreda. A Characterization of Badly Approximable Functions. *Transactions of the American Mathematical Society*, 169:249–256, 1972.

[75] N. Rivara, P.B. Dickinson, and A.T. Shenton. Constrained Variance Control of Peak-Pressure-Position by Spark-Ignition Feedback for Multi-Cyclinder Control. *International Journal of Advanced Mechatronic Systems*, 1(4):242–250, June 2009.

[76] N. Rivara, P.B. Dickinson, and A.T. Shenton. A Neural Network Implementation of Peak Pressure Position Control by Ionization Current Feedback. *Journal of Dynamic Systems, Measurement, and Control*, 131(5), 2009.

[77] D. Sarason. Generalized Interpolation in H-infinity. *Transactions of the American Mathematical Society*, 127:179–203, 1967.

[78] A.H. Sayed and T. Kailath. A Survey of Spectral Factorization Methods. *Numerical Linear Algebra with Applications*, 8:467–496, 2001.

[79] A.H. Sayed, H. Lev-Ari, and T. Kailath. Fast Triangular Factorization of the Sum of Quasi-Toeplitz and Quasi-Hankel Matrices. *Linear Algebra Application*, 1992.

[80] Z. Shafiei. *Design and Analysis of Robust Control Systems by Frequency Response Methods*. PhD thesis, University of Liverpool, 1991.

[81] Z. Shafiei and A.T. Shenton. Theory and Application of H-infinity Disk Method. Technical report, University of Liverpool, 1990.

[82] A.T. Shenton. Automotive Applications of Control. In *UKAC International Control Conference*, 2006.

[83] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control : Analysis and Design*. Wiley, New York, 2005.

[84] J. Stewart. *Calculus*. Cengage Learning, 2009.

[85] R. L. Streit and A.H. Nuttall. A Note on the Semi-Infinite Programming Approach to Complex Approximation. *Mathematics of Computation*, 40:599–605, 1983.

[86] R.L. Streit. Algorithm 635 : An Algorithm for the Solution of Systems of Complex Linear Equations in the L-infinity Norm with Constraints on the Unknowns. *ACM Transactions on Mathematical Software*, 3:242–249, 1985.

[87] R.L. Streit. Solution of Systems of Complex Linear Equations in the L-infinity Norm with Constraints on the Unknowns. *SIAM Journal of Scientific and Statistic Computing*, 7(1):132–149, January 1986.

[88] R.L. Streit and A.H. Nuttall. Linear Chebyshev Complex Function Approximation and An Application to Beamforming. *J. Acoust. Soc. Amer.*, 72(1):181–190, 1982.

[89] L.N. Trefethen. Rational chebyshev approximation on the unit disk. *Journal of Numer. Math.*, 37:297–320, 1981.

[90] L.N. Trefethen. Matlab programs for cf approximation. *Approximation Theory V*, Academic Press:599–602, 1986.

[91] F. van Diggelen and K. Glover. A Hadamard Weighted Loop Shaping Design Procedure. In *Proceedings of the 31st Conference on Decision and Control*, 1992.

[92] M. van Diggelen and K. Glover. Element-by-element Weighted H-infinity-Frobenius and H-2 Norm Problems. In *In proceedings of the IEEE Conference on Decision and Control*, 1991.

[93] M. Vidyasagar. *Control System Synthesis : A Factorization Approach*. MIT Press, Cambridge, MA, 1985.

[94] C.P. Ward. *J-Spectral Factorization for Automotive Powertrain Controls*. PhD thesis, University of Liverpool, 2009.

[95] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, New York, 1949.

[96] N. Wiener and L. Masani. The Prediction Theory of Multivariate Stochastic Processes, Part i. *Acta Methematica*, 98:111–150, 1957.

[97] N. Wiener and L. Masani. The Prediction Theory of Multivariate Stochastic Processes, Part ii. *Acta Mathematica*, 99:93–137, 1958.

[98] G. Wilson. Factorization of the Covariance Generating Function of a Pure Moving Average Process. *SIAM Jounral on Numerical Analysis*, 6(1):1–7, March 1969.

[99] G.T. Wilson. The Factorization of Matricial Spectral Densities. *SIAM Jounral of Applied Mathematics*, 23(4):420–426, December 1972.

[100] D.C. Youla. On the Factorization of Rational Matrices. *IRE Trans. Information Theory*, IT-7:172–189, 1961.

[101] D.C. Youla., J. J. Bongiorno, and C.N. Lu. Single-Loop Feedback Stabilization of Linear Multivariable Dynamic Plants. *Automatica*, 10(2):159–173, 1974.

[102] D.C. Youla, H.A. Jabr, and J. J. Bongiorno. Modern Wiener-Hopf Design of Optimal Controller, Part ii : The Multivariable Case. *IEEE Transactions on Automatic Control*, AC-21(4):319–338, 1976.

[103] G. Zames. On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems Part i : Conditions Derived using Concepts of Loop Gain, Conicity and Positivity. *IEEE Transactioins on Automatic Control*, AC-11, 1966.

[104] G. Zames. Feedback and Optimal Sensitivity : Model Reference Transformations Multiplicative Seminorms and Approximate Inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320, 1981.

[105] Y. Zhang. Solving large-scale linear programs by interior-point methods under the matlab environment. Tr96-01, Department of Mathematics and Statistics, University of Maryland, 1995.

[106] S. Zhao. *Nonparametric Robust Control Methods for Powertrain Control*. PhD thesis, University of Liverpool, 2011.

[107] S. Zhao and A. Abass.and A.T. Shenton. Nonparametric Design of Robust Linear Controller and Their Experimental Application to Idle Control. In *11th International Conference on Control and Applications*, 2009.

[108] S. Zhao and A.T. Shenton. A Nonparametric Method for Mixed Sensitivity Frobenius norm H-infinity Controller Design. In *8th UKACC International Control Conference*, 2010.

[109] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1996.