



## City Research Online

### City, University of London Institutional Repository

---

**Citation:** Nandakumar, R. (2008). Robust control design using quantitative feedback theory. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

---

**Permanent repository link:** <https://openaccess.city.ac.uk/id/eprint/19631/>

**Link to published version:**

**Copyright:** City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

**Reuse:** Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

# ROBUST CONTROL DESIGN USING QUANTITATIVE FEEDBACK THEORY

By

Ramnath Nandakumar

A Dissertation submitted for the degree of  
Doctor of Philosophy

Supervisor: Dr. George Halikias

City University  
School of Engineering & Mathematical Sciences  
Control Engineering Research Centre  
Northampton Square,  
London, EC1V 0HB

November 10, 2008



# **IMAGING SERVICES NORTH**

Boston Spa, Wetherby  
West Yorkshire, LS23 7BQ  
[www.bl.uk](http://www.bl.uk)

**THE FOLLOWING HAVE BEEN REDACTED AT THE  
REQUEST OF THE UNIVERSITY**

**DO NOT SCAN APPENDIX D**

# Declaration

I Ramnath Nandakumar, hereby certify that this thesis been written by me, that it is the record of work carried out by me and that it has not been submitted for any other higher degree qualification.



# Contents

<b>Declaration</b>	<b>1</b>
<b>Notation</b>	<b>10</b>
<b>Abbreviations</b>	<b>11</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis outline . . . . .	7
1.2 Thesis Contribution . . . . .	9
1.3 Publications resulting from this work . . . . .	10
<b>2 Literature Survey</b>	<b>11</b>
<b>3 Design Background</b>	<b>17</b>
3.1 Nichols Plot . . . . .	18
3.1.1 M-Circle . . . . .	18
3.1.2 N-Circles . . . . .	21
3.1.3 Transfer of $M$ circles to Nichols Plot . . . . .	22
3.2 Stability . . . . .	26
3.2.1 Nyquist Stability Criterion . . . . .	26
3.2.2 Stability criterion on the Nichols Chart . . . . .	29
3.3 Uncertainty . . . . .	33
3.3.1 Parametric Uncertainty . . . . .	35
3.3.2 Frequency Domain . . . . .	36
<b>4 QFT Design</b>	<b>39</b>
4.1 Design Specification . . . . .	41
4.2 Uncertainty Templates . . . . .	45
4.2.1 U-Contour . . . . .	49
4.3 Tracking Bounds . . . . .	52

4.4	Disturbance Rejection . . . . .	56
4.5	Disturbance Bounds $B_D$ . . . . .	57
4.6	Loop-Shaping . . . . .	61
4.7	Pre-Filter . . . . .	65
4.8	Multivariable system . . . . .	67
<b>5</b>	<b>Loop-shaping</b>	<b>71</b>
5.1	Linear Programming . . . . .	72
5.2	Bode integral . . . . .	72
5.2.1	Derivation . . . . .	74
5.2.2	Approximate Discretisation of Bode's integral . . . . .	78
5.3	Analytic constraints . . . . .	81
5.4	Fixed-structure controllers . . . . .	82
5.4.1	Introduction . . . . .	82
5.4.2	Problem description . . . . .	83
5.4.3	Formulation of QFT constraints as frequency-dependent inequalities . . . . .	86
5.4.4	Optimisation algorithm . . . . .	88
5.4.5	Optimisation algorithm steps: . . . . .	101
5.4.6	Design example . . . . .	103
5.5	Conclusions . . . . .	106
<b>6</b>	<b>QFT Software Toolbox</b>	<b>107</b>
6.1	Introduction . . . . .	107
6.2	Template Generation . . . . .	108
6.2.1	Algorithm Pseudo-Code . . . . .	109
6.3	Convex Hull . . . . .	109
6.3.1	Algorithm Pseudo-Code . . . . .	111
6.4	Horowitz Bounds (Tracking) . . . . .	113
6.4.1	Algorithm Pseudo-Code . . . . .	113
6.5	Disturbance rejection bounds . . . . .	115
6.5.1	Algorithm Pseudo-Code . . . . .	116
6.6	High Frequency Contour ( $U$ -Contour) . . . . .	116
6.6.1	M-Circle . . . . .	117
6.6.2	Algorithm Pseudo-Code . . . . .	121
6.6.3	Algorithm Pseudo-Code . . . . .	123
6.7	Graphical Stability . . . . .	124

6.8	Stability criterion . . . . .	125
6.8.1	Nyquist Stability criterion . . . . .	125
6.8.2	Stability criterion on the Nichol's Chart . . . . .	125
6.9	Graphical design . . . . .	127
6.9.1	Nyquist Criterion . . . . .	127
6.9.2	Nichol's Criterion . . . . .	128
6.9.3	Algorithm Pseudo-Code . . . . .	128
6.10	Loop Shaping . . . . .	130
6.10.1	Manual loop-shaping tool . . . . .	131
6.10.2	Fixed-structure controller optimisation . . . . .	133
6.11	Pre-Filter . . . . .	136
6.12	Conclusions . . . . .	136
<b>7</b>	<b>Case study - Design of hydraulic actuator</b>	<b>137</b>
7.1	Introduction . . . . .	137
7.2	Background and General Design Objectives . . . . .	138
7.3	Modelling of Hydraulic actuator interacting with environment . . . . .	140
7.4	Uncertainty modelling and QFT Control Design . . . . .	148
7.5	Conclusions . . . . .	158
<b>8</b>	<b>Conclusions, Future Work</b>	<b>160</b>
8.1	Conclusions . . . . .	160
8.2	Limitations . . . . .	162
8.3	Future Work . . . . .	162
	<b>Bibliography</b>	<b>164</b>
<b>9</b>	<b>Appendix</b>	<b>I</b>
9.1	Appendix A . . . . .	I
9.2	Appendix B . . . . .	III
9.3	Appendix C . . . . .	VIII
9.3.1	M-file to demonstrate controller design using QFT method. . . . .	VIII
9.3.2	M-file to demonstrate Bode approximation. . . . .	XIV
9.3.3	The function used to generate the uncertainty templates . . . . .	XVI
9.3.4	M-file to generate the convex hull with the sub function . . . . .	XVI
9.3.5	M.file to calculate the Horowitz templates . . . . .	XIX
9.3.6	M.file to calculate disturbance bounds . . . . .	XXII

9.3.7	M-file used to construct $M$ -contours in the Nichol's chart . . . .	XXV
9.3.8	M-file to construct high-frequency bounds . . . . .	XXVI
9.3.9	M-file to check Nichols stability criterion . . . . .	XXVIII
9.3.10	M-file to design the optimum PID controller . . . . .	XXIX
9.3.11	M-File for graphical design of the controller . . . . .	XXXI
9.3.12	M-file to design the Pre-filter . . . . .	XLII
9.4	Appendix D . . . . .	XLIII

# List of Figures

1.1	<i>Feedback Configuration</i>	5
3.1	<i>M circle on Nyquist Plot</i>	19
3.2	<i>M circle for Case1, Case2 and Case3</i>	20
3.3	<i>M circle for <math>M &gt; 1</math></i>	22
3.4	<i>M circle when <math>M = 1</math></i>	23
3.5	<i>M circle on Nichols Plot for <math>M &gt; 1, M &lt; 1, M = 1</math></i>	23
3.6	<i>M circle when <math>M &lt; 1</math></i>	24
3.7	<i>Nichols Plot</i>	25
3.8	<i>S - Plane</i>	27
3.9	<i>Positive and Negative crossing, and stability lines <math>S_c</math> and <math>S_n</math> on the Nyquist plane and Nichols chart</i>	30
3.10	<i>Stability region shown on Nichols chart, Fig (a) A. One crossing B. No corssing, Fig (b) Several crossings</i>	31
3.11	<i>Plant with Additive fig(a) and Multiplicative fig(b) Uncertainty's</i>	35
3.12	<i>Parameter Uncertainty</i>	36
3.13	<i>Uncertainty Represented on Nyquist plot</i>	37
4.1	<i>Bounds on step response of the plant</i>	42
4.2	<i>Bounds on the magnitude and phase responses of the plant</i>	42
4.3	<i>Modified bounds on plant step response</i>	43
4.4	<i>Modified bounds on magnitude and phase responses</i>	44
4.5	<i>Parameter Uncertainty</i>	46
4.6	<i>Uncertainty Template</i>	46
4.7	<i>Nominal Open-loop and Uncertainty Templates</i>	48
4.8	<i>M-circle in Nyquist plane (<math>M &gt; 1</math>)</i>	49
4.9	<i>M-circle and <math>\mathcal{U}</math>-contour in Nichols chart (<math>M &gt; 1</math>)</i>	51
4.10	<i><math>\mathcal{U}</math>-contour</i>	52

4.11	<i>Tracking Bound placements on the Nichols chart using Uncertainty Templates . . . . .</i>	54
4.12	<i>Horowitz Bounds . . . . .</i>	55
4.13	<i>Step response for <math>M_d</math> . . . . .</i>	58
4.14	<i>Bode plots for <math>M_d</math> . . . . .</i>	58
4.15	<i>Disturbance bounds <math>B_D</math> . . . . .</i>	60
4.16	<i>Overall bounds . . . . .</i>	60
4.17	<i>Loop-shaping phase correction . . . . .</i>	64
4.18	<i>Final Loop-shaping . . . . .</i>	65
4.19	<i>Step response of the system with the Pre-filter and controller in cascade</i>	66
4.20	<i>Frequency response of the closed-loop system . . . . .</i>	67
4.21	<i>General Multivariable system . . . . .</i>	67
4.22	<i>Expanded Multivariable system . . . . .</i>	68
4.23	<i>General feedback control system . . . . .</i>	69
4.24	<i>Expanded Multivariable system . . . . .</i>	70
5.1	<i>Differentiability region . . . . .</i>	75
5.2	<i>Phase reconstruction for first transfer function . . . . .</i>	80
5.3	<i>Phase reconstruction for second transfer function . . . . .</i>	80
5.4	<i>Differentiability region . . . . .</i>	82
5.5	<i>Feedback Configuration . . . . .</i>	84
5.6	<i>Curve <math>N(\phi)</math> used to calculate closed-loop bandwidth . . . . .</i>	90
5.7	<i>Optimal Loop-shaping with PID controller . . . . .</i>	104
5.8	<i>Closed-loop system frequency response . . . . .</i>	105
5.9	<i>Step responses of plants . . . . .</i>	105
5.10	<i>Disturbance rejection in the time domain . . . . .</i>	106
6.1	<i>Nominal Open-loop and Uncertainty Templates . . . . .</i>	110
6.2	<i>Convex hull design . . . . .</i>	112
6.3	<i>Convex hull of the uncertainty template at six frequencies . . . . .</i>	113
6.4	<i><math>M</math> circle for Case1, Case2 and Case3 . . . . .</i>	118
6.5	<i>Phase regions on the Nichol's chart . . . . .</i>	121
6.6	<i><math>M</math> circle on Nichol's Plot for <math>M = 1.2</math>, <math>M = 1</math> and <math>M = 0.8</math> . . . . .</i>	123
6.7	<i>M-circle and <math>\mathcal{U}</math>-contour in Nichol's chart (<math>M &gt; 1</math>) . . . . .</i>	124
6.8	<i>Positive and Negative crossing, and stability lines <math>S_c</math> and <math>S_n</math> on the Nyquist plane and Nichol's chart . . . . .</i>	125
6.9	<i>Stability region given on a Nichol's chart . . . . .</i>	127

6.10	<i>Graphical test flowchart</i> . . . . .	129
6.11	Lead/Lag-network manual loop shaping tool . . . . .	132
6.12	Control Design with Lead/Lag Network . . . . .	133
6.13	Design with Optimal PID controller . . . . .	135
7.1	Diagram of the hydraulic actuator (based on [40]) . . . . .	140
7.2	Manipulator-sensor-environment . . . . .	145
7.3	Closed-loop design tracking specifications, upper and lower bounds . . .	150
7.4	Uncertainty template, fifth design frequency . . . . .	151
7.5	Uncertainty template, sixth design frequency . . . . .	152
7.6	Uncertainty template, seventh design frequency . . . . .	153
7.7	Nominal-plant frequency response, Horowitz templates and $\mathcal{U}$ -contour .	154
7.8	Nominal open-loop response with $PDD^2$ -optimal controller . . . . .	155
7.9	Nominal open-loop response with modified $PDD^2$ -optimal controller . .	156
7.10	Shaped open-loop response and uncertainty templates . . . . .	157
7.11	Closed-loop frequency responses - No pre-filter . . . . .	158
7.12	Closed-loop frequency responses - With pre-filter . . . . .	159
7.13	Closed-loop step responses - With pre-filter . . . . .	159
9.1	Feedback Configuration . . . . .	XLV
9.2	M-circle in Nyquist plane ( $M > 1$ ) . . . . .	XLVIII
9.3	M-circle and $\mathcal{U}$ -contour in Nichols chart ( $M > 1$ ) . . . . .	XLIX

# List of Tables

1. *Magnitude and Phase values of corresponding frequencies*
2. *Values of  $\delta(j\omega_i)$  over 8 design frequencies*
3. *Open-loop for robust disturbance rejection*
4. *operating values and parameter range*
5. *Closed-loop specifications, Maximum and minimum gain*
6. *Target and achieved pre-filter gains*



# Notation

$\mathcal{R}, \mathcal{C}, \mathcal{N}$	Sets of real, complex and natural numbers
$\mathcal{R}(s)$	field of rational functions in $s$ with real coefficients
$\mathcal{R}[s]$	the set of real polynomials in the variable $s$
$\partial\mathcal{D}$	Boundary of set $\mathcal{D}$
$\mathcal{U}$ -contour	Closed contour in Nichols chart
$f_i^t(\phi)$	Horowitz templates
$f_i^d(\phi)$	Disturbance templates
$\mathcal{K}$	Family of all stabilising controllers
$\mathcal{S}$	Set of all stable closed-loop systems
$\mathcal{T}$	Set of all stable control sensitivity functions
$M_p$	Maximum peak
$t_p$	Peak time
$t_s$	Settling time
$t_r$	Rise time

Throughout this thesis matrix dynamical systems appear inside parenthesis so that they are distinguished from constant matrices which are denoted by square brackets.

# Abbreviations

<b>QFT</b>	Quantitative Feedback Theory
<b>PID</b>	Proportional Integral and Derivative Controller
<b>PI</b>	Performance Index
<b>LQR</b>	Linear Quadratic Regulator
<b>LQG</b>	Linear Quadratic Gaussian
<b>CAD</b>	Computer Aided Design
<b>LMI</b>	Linear matrix inequality
<b>LTl</b>	Linear time invariant
<b>MIMO</b>	Multiple input multiple output
<b>UHFB</b>	Universal High Frequency Bound
<b>LP</b>	Linear Programming
<b>RHP</b>	Right Hand Pole
<b>SISO</b>	Single input single output
<b>SSV</b>	Structured singular value
<b>SVD</b>	Singular value decomposition
<b>SIMO</b>	Single input multiple output

## Abstract

This thesis presents results in the area of robust control design using the Quantitative Feedback Theory (QFT) methodology. The thesis outlines the main philosophy and the various stages of this design approach and develops computational tools for carrying out a systematic design of uncertain feedback control systems in this framework using techniques of graphical design and computational geometry. Further, the thesis develops optimisation-based control design methods which can be carried out within a QFT computer-aided-design environment, with main emphasis on automatic loop-shaping. Two main design algorithms are proposed. The first involves the robust design of uncertain systems using fixed-structure controllers (PID, phase-lead/lag etc) which are widely used in practice. The second method is based on linear programming, and attempts to design the optimal controller in the frequency domain, subject to robust stability and performance specifications, augmented by additional realisability constraints based on the Bode gain/phase integral relationship. The proposed methods are tested via simple design examples and a detailed case-study involving the design of a non-linear hydraulic actuator. Simulation results of the closed-loop system demonstrate the applicability of the proposed techniques for the effective design of uncertain complex systems.

# Chapter 1

## Introduction

In this work we develop robust control design techniques based on a methodology known in the literature as *Quantitative Feedback Theory (QFT)*. This method, first introduced by I. Horowitz [28] in the early 70's, applies to the control design of dynamic systems which are subject (to potentially large) uncertainty in plant dynamics. QFT is a systematic procedure for designing systems of this type, and can guarantee “worst-case” performance and stability properties to the designed closed-loop system, in the sense that these properties apply over the whole family of models describing the uncertain plant dynamics. This characteristic makes QFT a *robust* control design methodology.

QFT is essentially a *loop-shaping* design procedure. The design objectives are typically formulated in terms of bounds on the closed-loop frequency response characteristics, which in turn can be translated to constraints on the open-loop frequency responses of the plant. The design requires shaping the open-loop frequency-response characteristics, so that these constraints are satisfied (over all frequencies). If the constraints are feasible, an appropriate optimality criterion is typically introduced, which allows the designer to select the “best” design by choosing the most appropriate feedback control scheme. In this sense, QFT is also an *optimal* control-design methodology, since loop-shaping is normally performed with this optimality criterion in mind. In this chapter the QFT design methodology is outlined for uncertain, linear, single-input single-output systems subject to typical stability and performance requirements. Before this description, however, we introduce the main characteristics of the method by describing two important control-design methodologies, *optimal* and *robust* control.

*Optimal Control:* Optimal control is a design approach which aims at getting the *best possible* performance out of a plant. This objective is normally achieved by optimising a mathematical expression which incorporates all aspects of the design which are deemed to be important, e.g. stability margins, performance objectives, etc. This expression is called the *performance index* (P.I.). The optimal controller is obtained by solving an optimisation problem, typically involving a number of constraints. One of the most successful optimal control methodologies was LQR/LQG (Linear Quadratic Regulator/Linear Quadratic Gaussian). In this method one assumes that the exogenous inputs (disturbances and noise signals) entering the system can be modelled as *coloured* or *white* noise signals, and minimises a quadratic performance index involving the rms average power of the regulated variables, which typically include a linear combination of the state variables and the control signals. When all state variables are available for feedback, the optimal solution consists of an optimal state-feedback matrix which is calculated by solving an Algebraic Riccati Equation. When a number of (noisy) output signals (other than states) are available, the optimal solution is obtained via the separation principle and consists of an optimal state estimator (Kalman filter) combined with the optimal state feedback obtained using the LQR procedure. Although the LQR design has excellent robust stability properties (good guaranteed gain and phase margins), these are typically lost when a Kalman filter is employed. An optimal controller method which takes into account explicitly robust stability and performance requirements is  $H_\infty$  - optimal control which was developed in the last two decades. This method shares many aspects of its philosophy with QFT; however, the performance index which is optimised is formulated in terms of the infinity norm of the closed-loop transfer functions; this is in contrast to QFT design which optimises the open-loop frequency response characteristics similarly to classical control. Despite these differences, it is argued in this thesis that the two methods can fruitfully complement one another.

*Robust Control:* During the control design process it is typically assumed that the plant is represented by an linear time-invariant (LTI) model, typically obtained by linearising a non-linear process around a fixed equilibrium point. In practice, the plant is subjected to various changing conditions in its interactions with its environment which tend to move the plant's set point, and as a result the linearised model will also

change. Additional sources of uncertainty may be due to factors like wear and tear due to aging of components or due to changing environmental conditions. The plant may also encounter unaccounted external factors like disturbances, or its mathematical model may have errors arising due to lack of knowledge of the exact values of certain of its parameters by the designer, approximations made in the modelling process itself due to simplicity requirements, or inconsistencies in the transducers used to measure various signals used for feedback. A control system which is capable to accommodate the effects caused by all these uncertainty factors is called *robust*. In general, when it can be shown that the controller design is stable for all plants within a specified family which contains all possible sources of uncertainty (“model - uncertainty set”), then we say that the controller provides *robust stability*. Apart from robust stability, we are also interested in *robust performance*, i.e. the performance of the design should not degrade excessively when plant uncertainty is taken into account and control signals should be kept within realistic level/rate bounds. Typically, robust design methods (including QFT) employ optimality criteria of a “minimax” type, i.e. they optimise the design for the “worst-case” situation (e.g. signal, parameter) that can occur among those allowed by the corresponding uncertainty model. This implies that “robust” control methods can be conservative, and care must be taken to describe the uncertainty set as accurately as possible.

*Quantitative Feedback Theory* is a systematic robust control design methodology for systems subject to large parametric or unstructured uncertainty. QFT is a graphical loop-shaping procedure, traditionally carried out on the Nichol’s chart, which can be used for the control design of either SISO or MIMO uncertain systems, including non-linear and time-varying models [17, 28, 57, 16]. Relative to other robust-control design methodologies, QFT offers a number of advantages, apart from its utilisation of classical control-design techniques. These include: (i) The ability to assess quantitatively the “cost of feedback” [29], (ii) the ability to take into account phase information in the design process (this is ignored in many norm-based approaches, e.g.  $\mathcal{H}_\infty$  optimal control which is based on singular values), and (iii) the ability to provide “transparency” in the design, i.e. clear tradeoff criteria between controller complexity and the feasibility of the design objectives. Note that (iii) implies in practice that QFT often results in simple controllers which are easy to implement.

The QFT design procedure is based on the two-degree of freedom feedback configuration shown in Figure 1.1. In this diagram  $G(p, s)$  denotes the uncertain plant, while  $K(s)$  and  $F(s)$  denote the feedback compensator and pre-filter, respectively, which are to be designed. Note that model uncertainty is described by the  $r$ -parameter vector  $p \in \mathcal{P} \subseteq \mathbb{R}^r$  taking values in the set  $\mathcal{P}$ ; it is further assumed that  $G(p, s)$  has the same number of RHP poles for all  $p \in \mathcal{P}$ . Translating the uncertainty into the frequency domain, gives rise to the plant's "uncertainty templates" which are the sets:

$$\mathcal{G}_\omega = \{G(p, j\omega) : p \in \mathcal{P}\}$$

For each fixed frequency  $\omega$ ,  $\mathcal{G}_\omega$  defines a "fuzzy region" on the Nichol's chart which describes the uncertainty of the plant at frequency  $\omega$  in terms of magnitude (in dB's) and phase (in degrees). For design purposes, we construct  $N$  uncertainty templates corresponding to a discrete set of frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$  chosen to cover adequately the system's bandwidth.

The robust performance objectives of the design include good tracking of reference input  $r(s)$  and good attenuation of the disturbance signal  $d(s)$  entering at the system's output, despite the presence of uncertainty. The robust tracking objectives are captured by the set of inequalities:

$$\max_{p \in \mathcal{P}} \Delta \left| \frac{G(p, j\omega_i)K(j\omega_i)}{1 + G(p, j\omega_i)K(j\omega_i)} \right|_{\text{dB}} \leq \delta(\omega_i) := B_u(\omega_i)|_{\text{dB}} - B_l(\omega_i)|_{\text{dB}}$$

for each  $i = 1, 2, \dots, N$ , i.e. if, for each frequency  $\omega_i$ , the maximum variation in closed-loop gain as  $p \in \mathcal{P}$  does not exceed the maximum allowable spread in specifications  $\delta(\omega_i)$ , typically specified via two appropriate magnitude frequency responses  $B_u(\omega) = |B_u(j\omega)|$  and  $B_l(\omega) = |B_l(j\omega)|$ . Note that it is not necessary to bound the actual gain (but only the gain spread) since we assume that, (i) no uncertainty is associated with the feedback controller  $K(s)$ , and (ii) the pre-filter  $F(s)$  can provide arbitrary scaling to the closed-loop gain.

The robust disturbance-rejection objective can be satisfied by bounding the sensitivity function, i.e. by imposing constraints of the form

$$\max_{p \in \mathcal{P}} \left| \frac{1}{1 + G(p, j\omega_i)K(j\omega_i)} \right| \leq D(\omega_i)$$

for a (subset) of the design frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$ . Again these are typically spec-

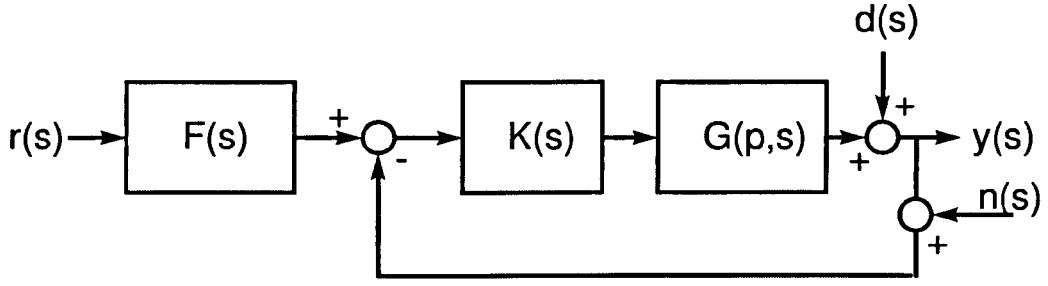


Figure 1.1: *Feedback Configuration*

ified via an appropriate magnitude frequency-response  $D(\omega) = |D(j\omega)|$ .

Robust stability is enforced by ensuring that: (i) no unstable pole-zero cancellations occur between the plant and the controller (for every  $p \in \mathcal{P}$ ), (ii) the nominal open-loop frequency response  $L_o(j\omega) = G(p_o, j\omega)K(j\omega)$  (defined for any  $p_o \in \mathcal{P}$ ) does not cross the  $-1$  point (i.e. the  $(-180^\circ, 0)$  point on the Nichols chart) and makes a total number of (anti-clockwise) encirclements around it equal to the number of unstable poles of  $L_o(s) = G(p_o, s)K(s)$ , and (iii) no (perturbed) open-loop response crosses the  $-1$  point, i.e.

$$-1 \notin \bigcup_{\omega \in \mathbb{R}} K(j\omega)\mathcal{G}_\omega$$

Note that condition (i) is automatically satisfied if  $K(s)$  is restricted to be stable and minimum-phase, while conditions (ii) and (iii) can be easily tested graphically [13, 12]. In practice, a more severe condition than (iii) is imposed: To establish a minimum amount of damping, it is required that the nominal open-loop frequency response does not penetrate a closed contour in the Nichol's chart ( $\mathcal{U}$ -contour); this is constructed from an appropriate  $M$ -circle and information about high-frequency uncertainty of the plant [17, 28].

The robust tracking and disturbance rejection objectives have been formulated as gain inequalities of the closed-loop transfer functions (sensitivity and complementary sensitivity) at the design frequencies. For the purposes of QFT design, these inequalities must be translated into constraints on the nominal open-loop response  $L_o(j\omega)$ . This procedure results into a number of contours (“Horowitz templates”  $f_i^t(\phi)$  and “disturbance-rejection templates”  $f_i^d(\phi)$ ) for each frequency  $\omega_i$ ,  $i = 1, 2, \dots, N$ ; these



are functions of the phase variable  $\phi \in (-360^\circ, 0^\circ]$ . Thus, robust tracking is satisfied at frequency  $\omega_i$  if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$  where  $\arg L_o(j\omega_i) = \phi_i$ ; similarly, robust disturbance-rejection is attained at frequency  $\omega_i$  if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi)$ . The robust-performance templates (Horowitz and disturbance-rejection) can be easily constructed (within an arbitrary gain tolerance and for a discretised phase-grid) using a simple bisection algorithm.

Once the contours corresponding to the robust stability and performance specifications have been defined, the design proceeds via *loop-shaping*. First, an arbitrary nominal plant  $G_o(s) = G(p_o, s)$  is selected, corresponding to an *arbitrary*  $p \in \mathcal{P}$ . The open-loop frequency response characteristics of  $L_o(s) = KG_o(s)$  are then shaped so that the robust stability and performance specifications are satisfied. This procedure is typically carried out by the designer in a CAD environment on the Nichols chart and requires a significant trial-and-error element. If the QFT constraints can be met, the best design is considered to be the one that requires as "little gain as possible". This requirement is formulated rather vaguely at present, but will be made precise in the sequel. Clearly, the objective here is to avoid an "over-design" of the system, by using higher gains than necessary. There are two main reasons for this requirement:

- A high open-loop gain implies a wide closed-loop bandwidth for the "complementary sensitivity" functions:

$$T(p, s) = \frac{G(p, s)K(s)}{1 + G(p, s)K(s)}$$

Now note that the transfer function from a sensor noise input  $n(s)$  to the plant output  $y(s)$  is  $-T(p, s)$ . Assuming that the spectrum of the sensor noise input is sufficiently "wideband", the larger the bandwidth of  $T(p, s)$ , the more "noisy" the output signal will appear. Thus, to prevent a noisy output signal  $y(t)$  we need to restrict the bandwidth of  $T(p, s)$  and thus the system's open-loop gain.

- A more important reason for the open-loop gain and thus also the closed-loop bandwidth of  $T(p, s)$  is related to robust stability: For unstructured multiplicative perturbations, the robust stability margins are inversely proportional to  $|T(p, j\omega)|$ . Since high frequency unstructured perturbations are typically present in practical systems due to the loss of phase information at high frequencies, unmodelled high frequency dynamics, etc, it is always desirable to avoid exces-

sive closed-loop bandwidths which may cause instability. Note that unstructured high frequency dynamics cannot be accounted by parametric uncertainty typically used to describe uncertainty in the standard QFT framework, although it is possible to modify the approach to take it into consideration.

Due to these two reasons, an optimal QFT design should make use of the minimal amount of gain, i.e. just enough to meet the robust performance objectives. Thus, it is typically attempted to shape the open-loop frequency response of the system so that  $L(j\omega_i)$ ,  $i = 1, 2, \dots, N$  lies exactly on, or just above, the corresponding Horowitz templates. This typically requires a significant amount of skill on the part of the designer and may be too difficult to perform adequately using a trial-and-error procedure. In the present work, it is attempted to alleviate this difficulty by automating the loop-shaping design procedure via a number of optimisation algorithms. These will be described in full in subsequent chapters of the thesis.

When an appropriate feedback controller has been designed, such that all robust stability and performance constraints are satisfied, the QFT procedure is concluded by designing a pre-filter to satisfy reference signal tracking specifications. This is typically a scaling exercise and can be performed using either a manual or an optimisation-based technique. Finally, closed-loop simulations are typically performed to validate the adequacy of the designed control scheme.

## 1.1 Thesis outline

The first chapter of the thesis outlines the QFT design procedure and formally defines the problem in an optimisation framework. The related areas of *robust* and *optimal* control are also briefly reviewed. A brief literature survey of the work related to this project is included in Chapter 2, with particular emphasis to QFT-based control design methods.

Chapter 3 describes the basic background of the QFT approach, including its main motivation and the various analysis tools employed, with particular emphasis on those related to the description of model uncertainty and graphical stability tests.

In Chapter 4 the QFT design method is described in detail. A number of theoretical results related to the method are stated, along with various examples illustrating each step. Since it is common practice to carry out the design using the Nichol's chart, some related background material is also included. The emphasis of the exposition is on the development of novel CAD tools and algorithms which can assist the designer or automate altogether the more difficult steps of the procedure. A description of a number of such tools is included, and a number of known results are reformulated so that they can be checked automatically (e.g. robust stability conditions on Nichols chart).

Chapter 5 proposes two new controller design methods based on automatic loop-shaping techniques, and tests their effectiveness via design examples and simulations. In the first method, the robust performance objectives (arising from the Horowitz and  $\mathcal{U}$ -contours) are used to define the set of linear constraints of the linear programme. These are augmented by another set of constraints (realisability, analyticity) which ensure that the optimal frequency response is realisable by an LTI dynamic system corresponding to the feedback controller. The second algorithm is related to the design of simple controllers of a fixed structure (PID, phase lead/lag, second-order). Here, the optimisation is carried out over the set of controller parameters.

Chapter 6 describes in detail the various steps of the complete set of QFT design algorithms, leading to the implementation of the software tool. This includes routines related to graphical representation of various contours in the Nichols chart (e.g. stability regions, robust performance contours, M-circles, plant uncertainty templates, etc), routines drawn from computational geometry (e.g. convex hulls) and various routines implementing optimisation algorithms, mainly related to automatic loop shaping and controller design.

In chapter 7 a detailed case study of a non-linear hydraulic actuator modelling a real system is presented. The model is linearised around an operating point and the uncertainty in the nominal plant is quantified in terms of ten uncertain parameters assumed to vary independently over their corresponding ranges. The methods developed in the thesis are used to design a robust QFT controller which meets the defined robust sta-

bility and performance specifications. The design is validated via extensive simulations and direct comparison with designs reported in the literature [41] implemented on the real system.

The main conclusions of the thesis, together with an outline of the scope for future work are included in Chapter 8. Finally, the appendices appearing at the end of the report contain background material related to the project, derivations and proofs of various technical results, together with the Matlab software tool developed in this project.

## 1.2 Thesis Contribution

- The thesis develops novel optimization-based techniques for analysing and designing robust feedback controllers using the QFT method. Robust stability and performance bounds are represented as mathematical constraints and are subsequently used to formulate optimization problems and thus automate the loop-shaping design procedure. This replaces traditional design methods relying on manual loop-shaping which require expert knowledge from the designer and may result in sub-optimal control schemes.
- The emphasis throughout the work is to design simple-structure controllers which can be used in practical industrial control. This is consistent with the design philosophy of the QFT method and provides “transparency” to the design. Thus, more complex controllers are introduced only when simple structures are deemed to be inadequate in some sense.
- Part of the work described in the thesis develops a novel computer-based design environment for carrying out robust-control designs using the QFT method and for assessing their performance and stability properties. This is based on fusing together techniques from computational geometry and optimization. The environment can be used for the purposes of representing plant uncertainty, visualizing the problem constraints, carrying out manual and optimization-based feedback control designs and validating the properties of the resulting control schemes. All optimisation algorithms developed in this work have been successfully tested in this environment, along with a detailed case-study of a non-linear actuator.

### 1.3 Publications resulting from this work

- R. Nandakumar, G. Halikias and A. Zolotas, “An optimization algorithm for designing fixed-structure controllers using the QFT method”, 2002 IEEE International Symposium on Computer Aided Control Systems Design Proceedings, Glasgow, Schotland UK, September 18-20 2002.
- R. Nandakumar, G. Halikias and A. Zolotas, “A new Educational tool for robust c control design using the QFT method”, Proc. of the 42nd IEEE Conference on Decision and Control, Vol. 1, pp. 803-808, 9-12 December 2003.
- R. Nandakumar, G. Halikias and A. Zolotas, “Robust Control Design of a Hydraulic Actuator Using the QFT Method” Proc. European Control Conference 2007, no WeA10.4, Kos, Greece, July 2-5, 2007.
- G. Halikias, A. Zolotas and R. Nandakumar, “Design of optimal robust fixed-structure controllers using the quantitative feedback theory approach”, Proc. IMechE Part I: J. Systems and Control Engineering, 221(4), pp. 291-312, July 2007.

# Chapter 2

## Literature Survey

Quantitative feedback theory (QFT) was initially proposed by I.M.Horowitz in 1963 [28]. It is a design method for designing robust control systems for uncertain plants subject to structured, unstructured or mixed-type uncertainties. QFT was initially developed for SISO systems and later extended to the multivariable case. The main contribution of Horowitz's work was to formulate the loop-shaping problem for an unstable and/or non-minimum phase plant and show that this is equivalent to a problem involving a stable non-minimum phase plant, by appropriately re-defining its robust stability and performance bounds. Initially, the proposed design procedure did not involve unstable plants; however this procedure was later extended by Horowitz to the unstable case in [30]. In 1972 Horowitz and Sidi proposed a new procedure for carrying out the robust control feedback design, by shifting the plant's stability bounds [29]. Although this procedure was generally efficient it lacked a formal proof, an issue that was successfully addressed by Chen and Ballance [12]. The QFT technique was extended to SIMO systems by Breiner [5].

A major issue with controller design using the QFT method is enforcing stability to the closed-loop system. As in any control problem, system stability is the most important objective and has to be established before attempting to satisfy performance specifications. A method dealing with this issue was suggested by Cohen, Chait, Yaniv and Borghesani [43]. This is based on the well known stability criterion proposed by Nyquist. Since the Nichols chart is traditionally used for control design in QFT, the Nyquist stability criterion must be translated into the Nichols chart and, ideally, must be verified via an automatic graphical technique. Nyquist stability in the Nichols chart was further developed by Ballance and Chen [13], with particular emphasis on

non-minimum phase and unstable systems. Their work resulted in a simple test which can be verified automatically via purely graphical means. This is especially useful for QFT design within a automated CAD environment and is used extensively in this work.

Although the main objective of the QFT design is to achieve robust stability, it is also important to satisfy robust performance specifications. In QFT these are typically imposed in the form of frequency-domain bounds on the sensitivity, complementary sensitivity and control-sensitivity functions, over a discrete frequency grid. The robust performance objectives are normally regarded as the constraints of an optimisation problem, the optimality criterion being formulated in terms of system over-design and controller complexity. Thus, the optimal loop-shaping of the open-loop characteristics becomes a significant aspect of the design, i.e. identifying the design (among all “feasible” designs) which achieves an “optimal” solution for the system. Design of robust controllers using QFT for plants with uncertainty was investigated by Jayasuriya and Zhao [32, 33]. Sidi [44, 45] and Horowitz and Sidi [30] presented a robust control design method for uncertain non-minimum phase plants with required closed-loop performance. No explicit optimisation problem was formulated; however their method gives the designer valuable insights into the tradeoff between closed-loop performance and bandwidth limitation [29]. A solution to the control problem is achieved during the loop-shaping stage of the procedure, using geometric contours derived from the robust performance specifications and the description of plant uncertainty. This essentially involves a modification of the open-loop response of the system, required to lie in certain regions of the Nichols chart and specified by the geometric contours described above. [36] provides a method for designing non-minimum phase MIMO system.

In the procedure initially suggested by Horowitz, *manual loop-shaping* was used. This is performed via an iterative design procedure and requires considerable skill from the part of the designer. Manual loop-shaping is essentially a trial and error method. Consequently, if the graphical constraints seem infeasible, it is not possible to decide conclusively whether this is due to the simple structure of the controller used, to the limitation of the designer’s abilities, etc.

After the arrival of computers capable of carrying out complex calculations, a signif-

ificant amount of effort has been devoted in trying to develop *automatic loop-shaping* QFT procedures. Polygonal approximation of the uncertainty templates was employed by Longdon and East (1978). A similar computational technique applicable to non-rational transfer functions aiming to alleviate the construction of uncertainty templates of QFT was reported by Gautam and Natarai [22]. Template generation using parameter discretisation methods suffers from the “curse of dimensionality”. As a result in most problems the designer is forced to trade-off between choosing a coarse plant grid to minimize the computational burden versus a fine grid to maintain highly accurate robustness specifications. An attempt to alleviate this problem using methods not relying on gridding is proposed in [4].

Another fundamental problem of QFT involves the design of the feedback controller satisfying a discrete set of robust stability and performance specifications. Yaniv and Chait [61] proposed a design method using quadratic inequalities, which applies both to continuous and discrete-time systems. A technique based on including a measure of unstructured uncertainty, the amount of which is dictated by the circle criterion, was proposed by Wang [55]. An efficient design technique was first obtained by Gera and Horowitz [24]. An automatic loop-shaping algorithm using convex optimisation which optimises the location of the zeros of the controller was proposed by Chait [9]. This has the clear limitation that the denominator of the controller must be specified in advance. Certain ad hoc rules for this task have been proposed in [9], but these require significant skill and experience from the part of the designer. The single-loop feedback design technique by Horowitz and Sidi [29], matches sensitivity as well as robustness specification for the *exact* amount of model uncertainty, and its criterion for a good design is the high-frequency gain originating by the ideal Bode characteristic of a loop transmission function [2]. An adaptive algorithm to modify the controller’s parameters by reducing the effects of plant uncertainty without affecting closed-loop performance was proposed by Yaniv [56]. Gutman [26] developed an algorithm to identify the reduced plant uncertainty.

The common grounds between QFT and  $H_\infty$ -optimal control were explored by Theodor and Shaked [48], resulting in a combination of both QFT and  $H_\infty$  methods, aiming at designing robust  $H_\infty$  controllers with almost no over-design. A similar method for



reconciling QFT and robust multivariable control was proposed by [48] using an approximation technique. Using this method it was shown that Nichols chart robustness bounds can be calculated without gridding of the uncertainty set and instead can be approximately solved using standard tools of robust multivariable control. Further connections between QFT and modern robust control were established by Lee, Chait and Steinbuch [37]. In this work it is argued that the integration of optimal control synthesis and manual tuning in QFT design environment enables design of controllers with levels of performance that surpasses what can be achieved using only a single technique. A constructive example is used to demonstrate that QFTs open-loop tuning can be more transparent than tuning closed-loop weights, as in modern robust control. Another approach aiming to develop a design methodology by utilising the best features from both modern robust control and QFT is proposed in [1]. In contrast to QFT, modern robust control typically results in high-order (observer-based) controllers. In the above cited work the authors characterise a class of second-order three-parameter controllers (including PID and lead/lag compensators) satisfying given  $\mathcal{H}_\infty$  norm closed-loop specifications using simple geometric considerations. An example illustrating the method is applied to the design of a PID controller in the case of bounded sensitivity specifications. These results were extended in [35] to the problem of obtaining the complete set of PID parameters that attains prescribed gain and phase margins.

in [50] the authors review modern QFT design in the light of modern robust control, motivated by the desire to develop a more rigorous treatment of non-minimum phase systems and/or plants characterised by mixed parametric and non-parametric uncertainty models. In this “new” approach traditional QFT robust performance and stability bounds (Horowitz templates) are replaced by sensitivity function bounds. A modified sensitivity-based QFT formulation is proposed in which limitations on the choice of nominal plant are made transparent; this formulation results in open loop design bounds which are equivalent to the traditional QFT problem at zero phase angle, while over-bounding them elsewhere. This formulation is also shown to meet the same necessary condition for Bode feasibility as traditional QFT. A gradient-based formulation for obtaining sensitivity QFT performance bounds is reported in [49] leading to constrained optimisation methods for determining controller parameters. Topological

aspects of QFT-based methodologies are reported by [51], who also show that QFT design can be formulated as a “strong”  $\mathcal{H}_\infty$  optimisation problem.

Thompson and Nwokah [51] developed an algorithm for shaping minimum-gain controllers. A more recent trend in automatic loop-shaping involves the use of *convex optimisation* methods. Bryant and Halikias [6] introduced a design procedure based on linear-programming. Another method of optimal loop-shaping involving simple fixed-structure controllers was proposed by Zolotas and Halikias [63]. The last two methods (linear programming approach and fixed-structure controller optimisation) are developed further in this work. Gain-bandwidth optimisation methods of PID controllers in the context of QFT design are also developed in [53]. [53] also describes a constrained optimisation method aimed at reducing the excess gain-bandwidth of an initial control design thereby improving its performance, while robustness can be incorporated in the design if the parameter bounds are suitably specified.

A two-step approach for automatic QFT closed-loop design is proposed in [14]. Automatic loop shaping of low-order QFT controllers by non-iterative methods designed in an open-loop method were proposed in [59] and [60]. Linear programming optimisation techniques for solving the same problem are reported in [10]. It is argued that the proposed method outperforms alternative automated loop-shaping techniques based on convex optimisation, as QFT bounds are typically non-convex; over-bounding QFT bounds by convex sets can thus be strongly conservative.

QFT was initially developed as a SISO design methodology, although extensions to the multivariable case are possible via a technique which decomposes the problem to a number of independent MISO designs by assuming a diagonal feedback controller see [38]. Recent developments in this area include the work of [34] using a sequential approach (closure of “one-loop at a time”), [11] via a pseudo-diagonalization technique combined with diagonal-dominance methods and [39, 20] which relies on a non-sequential methodology. Another approach suggested by [21], to model MIMO system involves tracking error specifications. This method treats effects of uncertainty as output disturbances.

QFT is essentially a graphical design methodology and an important reason for

its relative success over recent years is that it develops classical control concepts in a natural and transparent way. As the complexity of the method, however, is considerably higher than classical control, the development of effective graphical user-friendly software tools are essential for its applicability. The standard commercial Matlab tool developed recently [3] is sufficiently versatile for the effective design of SISO systems with moderate complexity. It is important, however, for the success of the method that QFT-based software does not lag behind theory developments in the field. Open-source QFT tools applying recent QFT techniques are reported in [31] and [42]. In this work, a new Matlab-based toolbox has been developed based on the techniques reported in this thesis which, it is hoped, can make the QFT design technique accessible to the wider control community and serve as a test-bed for the implementation of novel techniques and algorithms.

# Chapter 3

## Design Background

In this chapter we provide the necessary background information about the various techniques used in this thesis for the design of robust controllers using the QFT method. The main platform for the QFT design is the Nichols chart. The first section in this chapter deals with the procedure for the construction of various design contours on the Nichols chart, especially  $M$  and  $N$ -contours.  $M$  circles in the Nyquist diagram are used in classical design to define regions which must be avoided by the open-loop frequency response, in order to provide a minimum damping for the closed-loop system (i.e. good stability gain and phase margins, limits on the sensitivity function, etc). They can be thought of as regions imposing stronger requirements on closed-loop stability than Nyquist conditions specifying the encirclements of the critical point  $(-1)$ . In QFT  $M$  contours are important for two reasons: (i) They are used to define the “high-frequency  $U$ -contour” which imposes robust minimum damping bounds to the design in the high-frequency range, and (ii) They can be used to define the “Horowitz templates” which specify the minimum open-loop gain necessary to achieve the maximum allowable spread in tracking specifications despite the presence of plant uncertainty. Both these contours are described fully in the sequel.

One of the main tools used to check closed-loop stability of a control system is the Nyquist stability criterion. Since the QFT design is entirely carried out in the Nichols chart, it is essential to have a rigorous procedure to verify the stability of the system in this domain. Section 3.2 of this chapter explains the transformation of the Nyquist stability criterion into the Nichols chart. Since the main advantage of the QFT design methodology is its ability to deal with plants with high uncertainty, section 3.3 in this chapter gives a brief description of various types of model uncertainty encountered in

practice. In addition, stability tests based on the nominal plant are generalised to *robust* stability tests, which ensure closed-loop stability for certain quantifiable measures of plant uncertainty.

## 3.1 Nichols Plot

The Nichols chart is a rectangular coordinates plot of magnitude and phase. It was first introduced by Nathaniel Burgess Nichols, b. 1914.

The Nichols plot is a graph of the open-loop phase (in degrees) vs open-loop magnitude (in dB) with the addition of superimposed closed-loop constant magnitude and phase contours. The system's closed-loop frequency response characteristics may be easily determined from these super - imposed contours once the frequency response of the open-loop system has been displayed.

The  $M$  and  $N$  circles in the Nyquist diagram which indicate the corresponding closed-loop gain and phase properties of the feedback design transform into non-circular  $M$  and  $N$  contours on the Nichols chart. Thus, closed-loop information can be immediately obtained from the open-loop frequency response plot of the system. In particular, the gain and phase margin of the design can be derived by considering the points where the open-loop frequency response crosses the magnitude axis and phase axis, respectively.

### 3.1.1 M-Circle

A  $M$ -Circle in the Nyquist diagram is defined as the locus of all open-loop frequency response points which corresponds to a fixed closed-loop magnitude  $M$  [52]. Consider the closed-loop plant shown in figure 3.1, whose frequency response is

$$\mathbf{T}(j\omega) = \frac{\mathbf{G}(j\omega) \mathbf{K}(j\omega)}{1 + \mathbf{G}(j\omega) \mathbf{K}(j\omega)} \quad (3.1)$$

Here  $G(j\omega)$  and  $K(j\omega)$  denote the frequency response of the (nominal) plant and the controller, respectively.

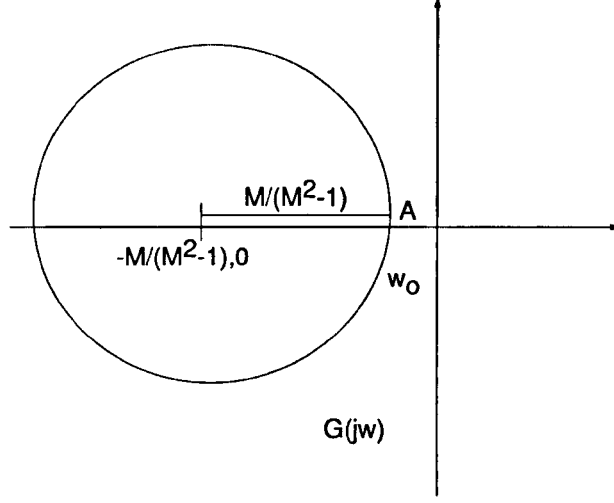


Figure 3.1:  $M$  circle on Nyquist Plot

The magnitude of the closed-loop system at frequency  $\omega$  is:

$$|T(j\omega)| = \left| \frac{G(j\omega) K(j\omega)}{1 + G(j\omega) K(j\omega)} \right| \quad (3.2)$$

We are interested in characterising the geometric locus of all points of the Nyquist plane at which  $|T(j\omega)| = M$  (constant). Let the open-loop frequency response of the plant at frequency  $\omega$  be  $G(j\omega) K(j\omega) = u + jv$ . Then:

$$M = \frac{|u + jv|}{|1 + u + jv|} = \frac{\sqrt{u^2 + v^2}}{\sqrt{(1+u)^2 + v^2}} \Rightarrow M^2 = \frac{u^2 + v^2}{(1+u)^2 + v^2}$$

Thus:

$$u^2(1 - M^2) + v^2(1 - M^2) - 2uM^2 = M^2 \quad (3.3)$$

or,

$$u^2 + v^2 - 2u \frac{M^2}{(1 - M^2)} = \frac{M^2}{(1 - M^2)} \quad (3.4)$$

By *completing the squares* in equation 3.4 we get:

$$\left( u + \frac{M^2}{M^2 - 1} \right)^2 + v^2 = \left( \frac{M}{M^2 - 1} \right)^2 \quad (3.5)$$

This equation represents a circle with its centre at  $\left( \frac{M^2}{M^2 - 1}, 0 \right)$  and radius  $\left| \frac{M}{M^2 - 1} \right|$  in the

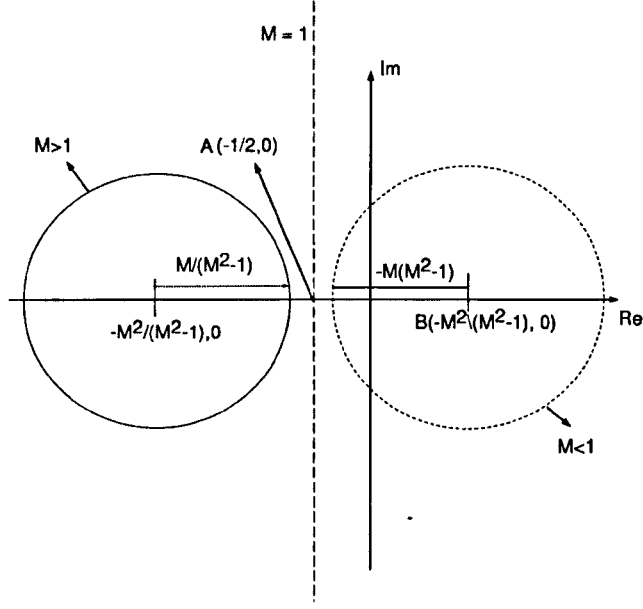


Figure 3.2:  $M$  circle for Case1, Case2 and Case3

polar (Nyquist) plane.

From equation 3.5, it can be noticed that the position of the  $M$  circle on the Nyquist plot varies with the value of  $M$ . This gives rise to three special cases which are considered in detail next:

### Case 1

The first case is  $M > 1$ . Since the centre of the circle is at  $\left(-\frac{M^2}{M^2-1}, 0\right)$ , we have:

$$\frac{-M^2}{M^2-1} < 1$$

Thus the centre of the circle in this case lies on the negative real axis. A typical plot is shown in figure 3.2. It is also easy to show that the  $M$  - circle lies to the left of the vertical line  $A$  through the co-ordinates  $\left(-\frac{1}{2}, 0\right)$  in this case.

## Case 2

The second case is when  $M = 1$ . Substituting  $M = 1$  in equation 3.3, the coordinates of the point A are  $(-\frac{1}{2}, 0)$ . In this case the plot is the vertical straight line  $A(-\frac{1}{2}, 0)$  as shown in the figure 3.2.

## Case 3

The last case is  $M < 1$ . Since:

$$\frac{-M^2}{M^2 - 1} > 1$$

the centre of the circle will lie on the right half of the plot as shown in figure 3.2, and the co-ordinates of the point B are  $(\frac{-M^2}{M^2 - 1})$ .

### 3.1.2 N-Circles

While the closed-loop gain of the plant can be obtained from the  $M$  circle on the Nyquist plane, the closed-loop phase information is provided by the  $N$  circles. Let  $N$  be a constant angle, and let  $K(j\omega) = u + jv$  at an arbitrary frequency  $\omega$ . Then the phase of the closed-loop system  $T(j\omega)$  is:

$$\arg T(j\omega) = \arg K(j\omega) - \arg(1 + K(j\omega)) \quad (3.6)$$

so that

$$\arctan(N) = \arctan\left(\frac{\frac{v}{u} - \frac{v}{1+u}}{1 + (\frac{v}{u})(\frac{v}{1+u})}\right)$$

By rearranging the above equation we get:

$$N = \frac{v}{u^2 + u + v^2}$$

which implies that

$$\left(u + \frac{1}{2}\right)^2 + \left(v - \frac{1}{2N}\right)^2 = \frac{1}{4} \left(\frac{N^2 + 1}{N^2}\right) \quad (3.7)$$

Equation 3.7 is an equation of a circle with centre at  $(-\frac{1}{2}, \frac{1}{2N})$  and radius  $\frac{1}{2N}\sqrt{N^2 + 1}$ .

The intersection of the open-loop locus of  $G(j\omega)$  on the  $N$  circle gives the phase of the closed-loop at the frequency corresponding to the point of intersection.



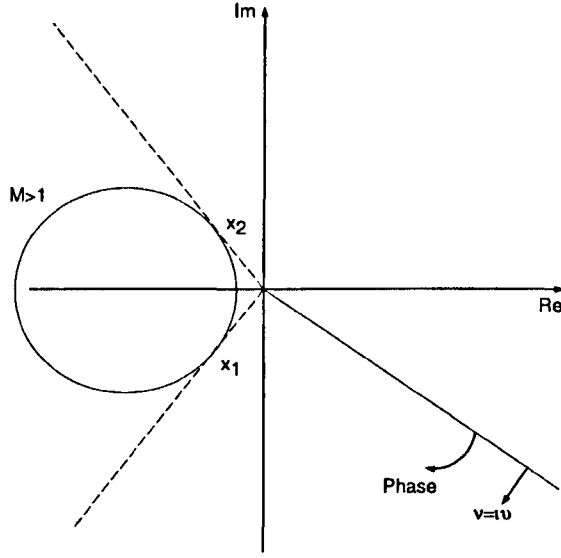


Figure 3.3:  $M$  circle for  $M > 1$

### 3.1.3 Transfer of $M$ circles to Nichols Plot

Based on the value of  $M$ , the pattern of the  $M$  circles on the Nichols chart can be obtained for the three cases outlined in the previous section. To transfer the  $M$  circles to the Nichols chart, consider a straight line (constant phase) through the origin described by the equation

$$v = lu \quad (3.8)$$

where  $u$  and  $v$  denote the real and imaginary parts of the open-loop frequency response respectively. To transfer the  $M$ -circles to the Nichols chart we need to solve the two equations ( $M$ -circle and straight line) simultaneously.

#### Case 1

When  $M > 1$  the circle will lie to the left or the negative half of the Nyquist plot as shown in figure 3.3.

From figure 3.3 it can be noticed that as the fixed phase line rotates through  $360^\circ$  it can be the tangent of the circle on two occasions, at point  $x_1$  and  $x_2$ . Thus the  $M$  circle on the Nichols plot will be defined only for the phase range:

$$-180^\circ - \sin^{-1} \left( \frac{1}{M} \right) \leq \phi \leq -180^\circ + \sin^{-1} \left( \frac{1}{M} \right)$$

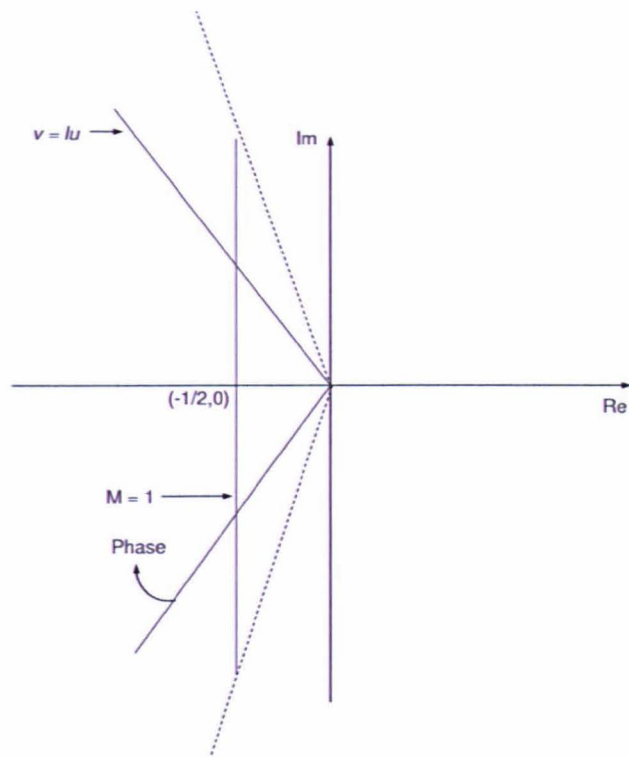


Figure 3.4:  $M$  circle when  $M = 1$

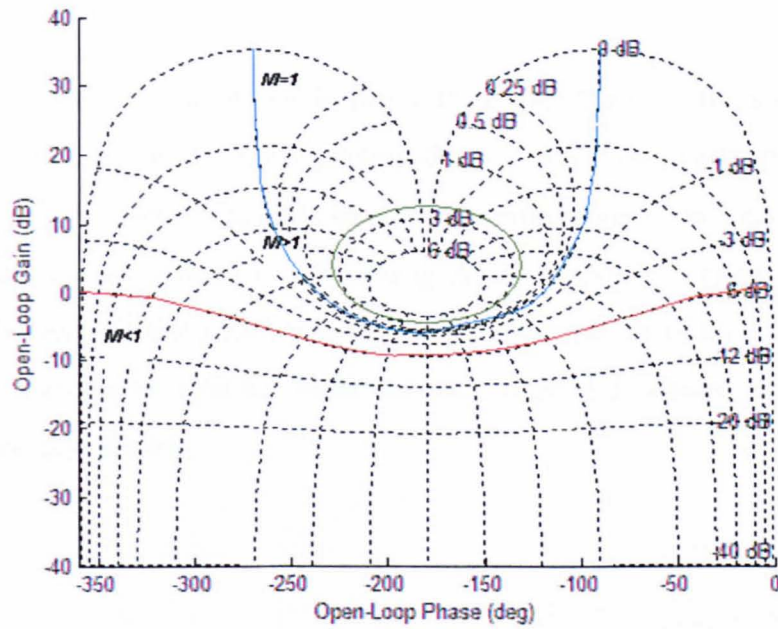


Figure 3.5:  $M$  circle on Nichols Plot for  $M > 1$ ,  $M < 1$ ,  $M = 1$

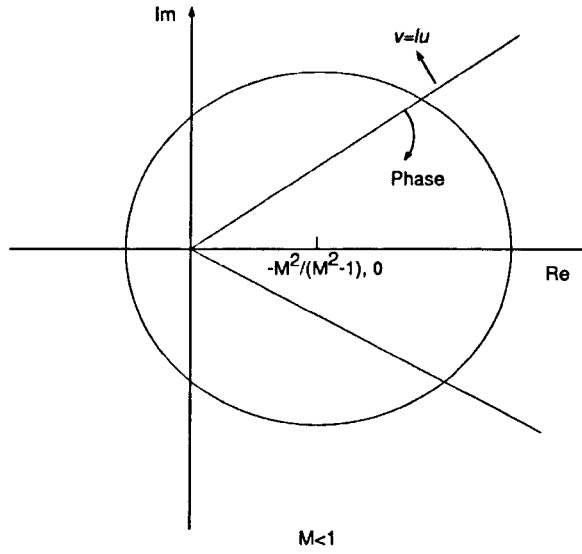


Figure 3.6:  $M$  circle when  $M < 1$

For any  $\phi$  (strictly) inside this interval there are exactly two (positive) solutions to the system of simultaneous equations corresponding to the  $M$  - circle and the straight line. It follows that the  $M$  circle on the Nichols chart in this case is a closed contour as shown in the figure 3.5.

## Case 2

As shown in figure 3.4 the  $M$  circle is a vertical line through the point  $(-\frac{1}{2}, 0)$ . Thus the closed-loop magnitude will tend to infinity when the open-loop phase approaches  $-270^\circ$  and  $-90^\circ$ . Clearly, in this case the simultaneous equations have one positive and one negative solution, the later being discarded since it cannot represent a gain variable. Converting this plot to Nichols chart corresponds to an open contour defined for phases in the open interval  $-270^\circ$  to  $-90^\circ$  only and tending to infinity as we approach these two phases.

## Case 3

Since in this case  $\frac{M^2}{M^2-1} < \frac{M}{M^2-1}$  the  $M$  circle in the Nyquist diagram encloses the origin. Thus the constant phase line  $v = lu$  will cross the circle at two distinct points which means that on solving the system of equations corresponding to the circle and

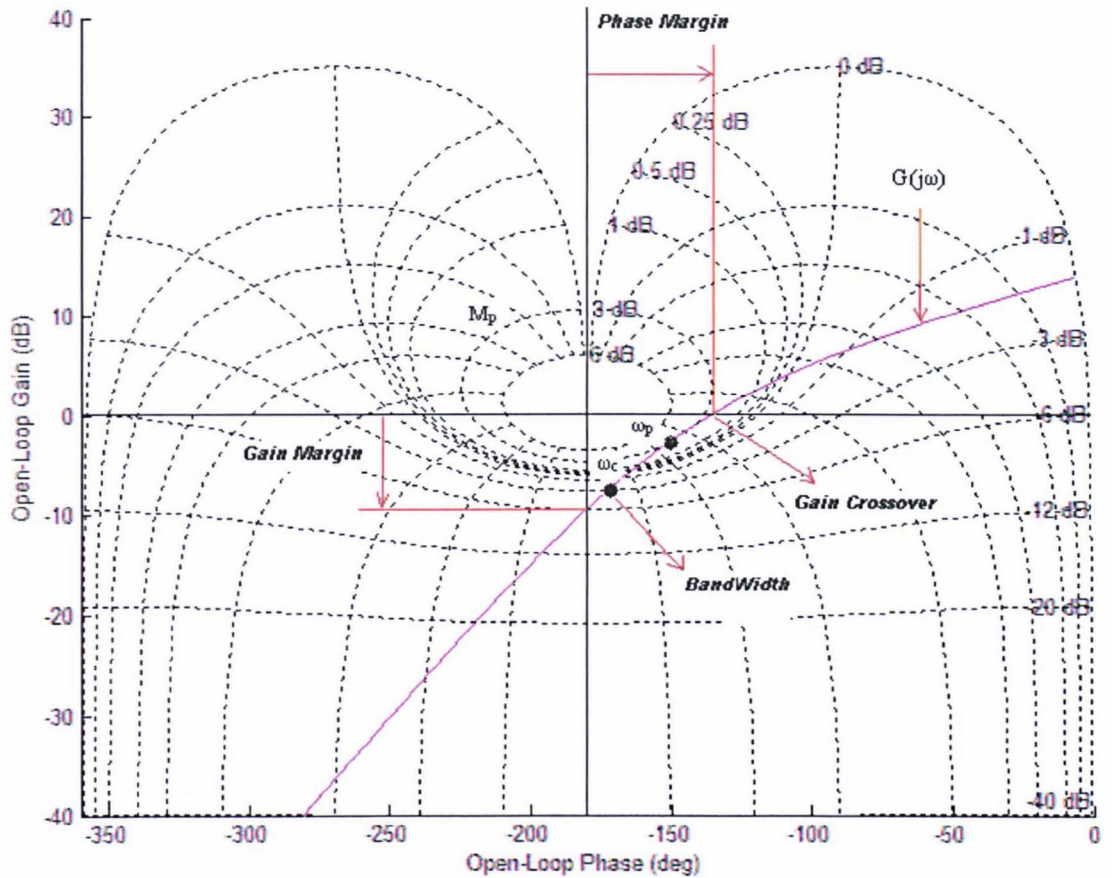


Figure 3.7: *Nichols Plot*

the straight line we always get two distinct positive solutions for  $x_1$  and  $x_2$ , i.e.  $x_1 \neq x_2$ . Thus the plot of  $M$ -circles on the Nichols chart will be an open-ended contour as shown in the figure 3..

For an open-loop stable system  $G(s)$ , the maximum magnitude ratio  $M_P$  is obtained by finding the largest  $M$  contour which touches, but does not cross the  $G(j\omega)$  locus.  $M_P$  is a useful design parameter since it indicates the maximum (“resonance”) peak of the closed-loop magnitude frequency response, or indirectly the “minimum damping” of the system. The frequency corresponding to  $M_p$   $\omega_p$  say, is not a direct reading, but can be easily obtained by interpolating the frequency-response  $G(j\omega)$  data points. The bandwidth of the closed-loop system is similarly found from the intersection of the  $G(j\omega)$  locus with the  $-3dB$  contour. Since all  $M$  - circles on the Nyquist plane are symmetric with respect to the negative real axis the corresponding contours in the

Nichols chart will be symmetric relative to the  $\phi = -180^\circ$  line.

## 3.2 Stability

A stable system is an absolute requirement for any control design. The system's stability conditions can be different in its open-loop and closed-loop state, and thus an open-loop stable system does not necessarily imply a closed-loop stable system. There are various methods available to check system stability. The most direct method is to identify the location of the closed-loop poles, which must all lie in the open left-half of the complex plane (i.e. have negative real parts). However, this test requires the full knowledge of the plant model and does not generalise easily to a test for "robust" stability, since this would typically require the calculation of the poles of an infinite number of systems (each corresponding to an uncertain plant). The most versatile stability test is based on the Nyquist stability criterion. This requires only the frequency response of the open-loop system (and the *number* of open-loop unstable poles) and generalises easily to produce robust-stability tests.

As mentioned earlier the QFT problem is traditionally formulated using the Nichols chart. Thus, in this section, the Nyquist stability criterion is re-formulated in this domain.

### 3.2.1 Nyquist Stability Criterion

The Nyquist stability criterion relates the total number of encirclements of the open-loop frequency response around the critical  $(-1)$  point to the number of the system's closed-loop poles that lie in the right half of the  $s$  - plane. The Nyquist stability criterion is based on the result from complex analysis, known in the literature as *Cauchy's principle of the argument*. [23, 8].

**Number of Poles and Zeros:** Let  $G(s)$  be analytic on the  $s$  - plane except at except at a finite number of points (namely, the poles of  $G(s)$ ) (figure 3.8). Then we may write  $G(s)$  using Laurent's series expansion as:

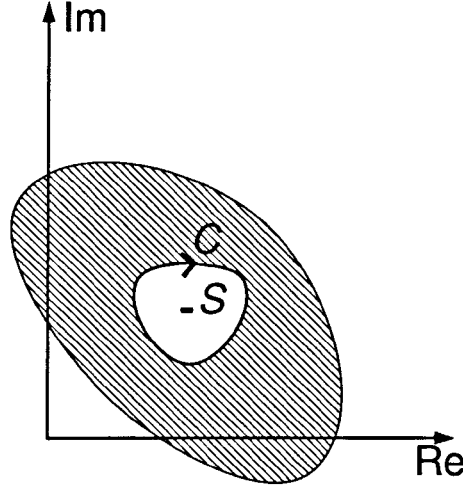


Figure 3.8:  $S$  - Plane

$$G(s) = \frac{A_{-n}}{(s - s_o)^n} + \dots + \frac{A_{-1}}{s - s_o} + B_o + B_1(s - s_o) + \dots \quad (3.9)$$

The coefficient  $A_{-1}$  is called the residue of  $G(s)$  at  $s_o$  and may be evaluated as

$$A_{-1} = \text{Res}[G(s); s_o] = \frac{1}{2\pi} \oint_C G(s) ds \quad (3.10)$$

where  $c$  denotes a closed arc within an analytic region centered at  $s_o$  that contains no other singularities.

*Theorem:* Let  $G(s)$  be an analytic function inside and on a closed contour  $C$  except for a finite number of poles inside  $C$ . Then, as we transverse  $C$  in the clockwise direction,

$$\frac{1}{2\pi} \oint \frac{G'(s)}{G(s)} ds = Z - P$$

Equivalently:

$$\frac{1}{2\pi} \oint d(\ln G) = Z - P \quad (3.11)$$

where  $Z$  is the number of zeros and  $P$  is the number of poles inside  $C$ .

*Proof:-* Let  $s_o$  be a zero of  $G$  with multiplicity  $k$ . Then in some neighbourhood of that point we may write  $G(s)$  as:

$$G(s) = k(s - s_o)^{k-1} f(s) + (s - s_o)^k f'(s) \quad (3.12)$$

where  $f(s)$  is analytic and  $f(s_o) \neq 0$ . If we differentiate equation 3.9, we get

$$G'(s) = k(s - s_o)^{k-1}f(s) + (s - s_o)^k f'(s)$$

from which it follows that:

$$\frac{G'(s)}{G(s)} = \frac{k}{s - s_o} + \frac{f'(s)}{f(s)}$$

Therefore  $\frac{G'(s)}{G(s)}$  has a pole  $s = s_o$  with residue  $k$ . This procedure is repeated for every zero. Hence the sum of the residues of  $\frac{G'(s)}{G(s)}$  is the number of zeros of  $G(s)$  inside  $C$ . If  $s_o$  is a pole with multiplicity  $l$ , we may write  $G(s)$  as

$$h(s) = (s - s_o)^l G(s) \quad (3.13)$$

where  $h(s)$  is an analytic function and  $h(s) \neq 0$ . From equation 3.13 we get:

$$G(s) = \frac{h(s)}{(s - s_o)^l}$$

Differentiating the above equation gives:

$$G'(s) = \frac{h'(s)}{(s - s_o)^l} - \frac{lh(s)}{(s - s_o)^{l+1}}$$

so that

$$\frac{G'(s)}{G(s)} = \frac{-l}{s - s_o} + \frac{h'(s)}{h(s)}$$

This analysis is repeated for every pole. It follows that the sum of the residues of  $\frac{G'(s)}{G(s)}$  at every pole of  $G(s)$  is equal to  $-P$ . Using equation 3.10

$$\frac{1}{2\pi j} \oint_C d(\ln G(s)) = Z - P$$

where  $d(\ln G(s))$  was substituted for  $\frac{G'(s)}{G(s)} ds$ . If we write  $G(s)$  in polar form then

$$\begin{aligned} \oint_{\Gamma} d(\ln G(s)) &= \oint_{\Gamma} d(\ln |G(s)| + j \arg(\ln G(s))) \\ &= \ln |G(s)| \Big|_{s=s_1}^{s=s_2} + j \arg G(s) \Big|_{s=s_1}^{s=s_2} \end{aligned}$$

Since  $\Gamma$  is a closed contour, the first term is zero, and the second term is  $s\pi$  times the net encirclements of the origin. Thus:

$$\frac{1}{2\pi j} \oint_{\Gamma} d(\ln G(s)) = Z - P$$

as required. □

**Remark:** If  $G(s)$  is stable, then its unity feedback closed-loop system is also stable if and only if, the Nyquist contour does not encircle the  $(-1, 0)$  point. If  $G(s)$  has  $P$  poles in the right-half of the  $s$ -plane, then the number of counter-clockwise encirclements of the  $(-1, 0)$  point must be equal to  $P$  for the corresponding closed-loop system to be stable. [58].

### 3.2.2 Stability criterion on the Nichols Chart

Since QFT design is carried out on the Nichols plane, it is sensible to have a criterion to specify stability of a system directly in this plane. A version of Nyquist stability criterion on the Nichols chart was first developed in [43]; this was further improved in [12].

The stability criterion on the Nichols chart is a re-formulation of the Nyquist stability criterion. Stability of a system in the Nyquist plane is mainly based in the *Zero exclusion theorem*, i.e. for a system to be stable the condition  $1 + L(j\omega) \neq 0$  should hold, and the net encirclements of the critical point  $-1$  should be zero i.e.  $N = P - Z$  where  $P$  is the number of poles and  $Z$  is the number of zeros of the system. In the Nyquist plot the direction of the system response produced by an unstable pole is in the *anti-clockwise* direction in the left half of the Nyquist - plane and this is called *Negative crossing* of the stability line  $S_c$ . The response produced by a zero in the system is in the *clock-wise* direction in the left half of the Nyquist - plane, and this is called *Positive crossing* of the stability line  $S_c$ . The stability line together with typical positive and negative crossings are shown in the figure 3.9.

Firstly the stability line  $S_c$  which contains the critical point  $-1$  in the Nyquist plane is translated onto the Nichols chart. The stability line in the Nyquist plane is given by

$$S_c =: \{(x, y) : y = 0, x < -1\} \quad (3.14)$$

$S_c$  is translated onto the Nichols chart using the relations  $Z = a + ib$ , where  $a = r \cos \theta$  and  $b = r \sin \theta$  (in this case  $a = -1$  and  $b = 0$ ), and so the stability line on the Nichols chart  $S_n$  is given by

$$S_n =: \{(\theta, r) : \theta = -180^\circ, r > 1\} dB \quad (3.15)$$



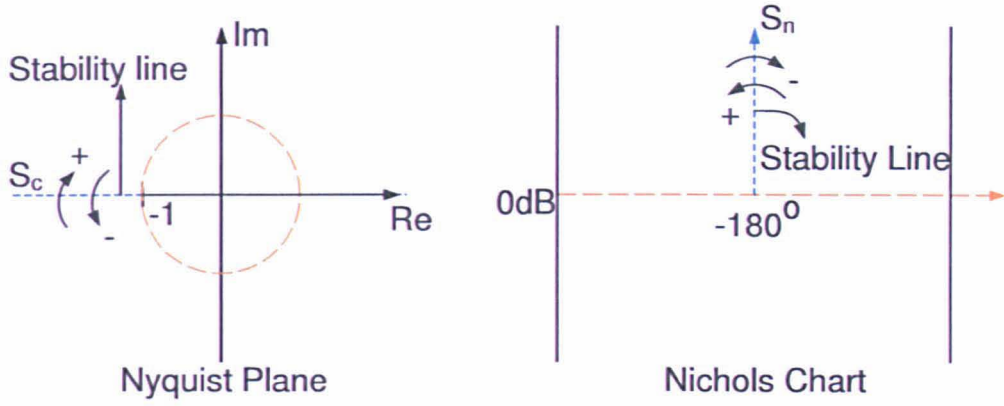


Figure 3.9: *Positive and Negative crossing, and stability lines  $S_c$  and  $S_n$  on the Nyquist plane and Nichols chart*

The stability lines on the Nyquist plane and on the Nichols chart are shown in figure 3.9. The stability criterion on the Nichols chart is formulated by applying the Nyquist stability criterion on the Nichols chart. The stability line in the Nichols chart is given by  $S_n$ , this line corresponds to the stability line  $S_c$  in the Nyquist chart as shown in figure 3.9. The stability analysis of a plant in the Nichols chart is based on the number of positive and negative crossing's of the stability line  $S_n$  by the open-loop response of the system.

To introduce the stability condition a stable system with  $n$  stable poles in equation 3.16 is considered. Let:

$$L(s) = \frac{N(s)}{D(s)} \quad (3.16)$$

The poles of the above system lie in the left half of the  $s$  - plane. The closed-loop stability of the system for various cases is guaranteed by the conditions below:

- For a system whose response lies above the line  $r = 0$  dB the open-loop response should pass through the line  $r = 0$  dB, in the range  $-180^\circ < 180^\circ$  to make the system response stable.
- As a direct consequence of the Nyquist stability criterion, the net positive and negative crossings of the stability line  $R_L =: \phi = -180^\circ, r = [0, \infty)$  and  $R_R =: \phi = 180^\circ, r = [0, \infty)$ , should be zero, i.e. if the system response **crosses** the stability lines  $R_L$  or  $R_R$ , then, in order for the system to be stable, the

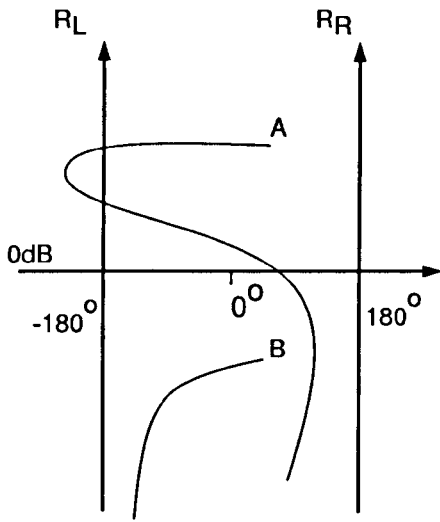


Fig a

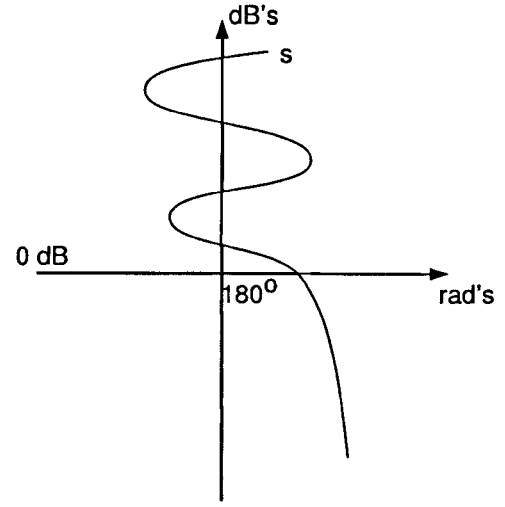


Fig b

Figure 3.10: *Stability region shown on Nichols chart, Fig (a) A. One crossing B. No crossing, Fig (b) Several crossings*

response should re-enter the stability region. Then the response should cross the line  $r = 0dB$  within the stability lines, thus satisfying the above condition.

- For a plant whose open-loop response starts below the line  $r = 0dB$ , the system is always stable.

### Nichols stability criterion for Unstable/Non minimum phase plants

The region of the open-loop response plot on the Nichols chart will be dictated by the presence of an unstable pole or zero in the plant. Depending on the number of unstable poles or poles the system response is shifted towards the right in multiples of  $n$ , where  $n$  is the pole-zero excess of the system. The stability of a system is checked using Nichols stability criterion by first transforming the unstable/non-minimum phase plant into a stable/minimum-phase plant, then shifting the robust stability bounds by a specific amount in the horizontal axis (phase value), and finally applying the stability criterion. In the process of converting the unstable/non-minimum phase plant into a stable/minimum-phase plant, the gain of the plant response should not be altered, in order to retain the plant characteristics. A procedure for finding the required phase shift for the robust bounds is given in [13, 25]. First consider an unstable/non minimum-

phase system factored as:

$$P_N(s) = \frac{N(s)\hat{Z}(-s)}{D(s)\hat{P}(-s)} \quad (3.17)$$

where  $\hat{Z}(-s)$  is an unstable polynomial containing all the non-minimum phase zeros of  $N(s)$ , i.e. all zeros in the right half of the  $s$  - plane; similarly  $\hat{P}(-s)$  contains all unstable poles, i.e. all plant poles in the right half of the  $s$ - plane. The general expression for robust stability margin for a plant in QFT design is given by:

$$\left| \frac{L_N(s)}{1 + L_N(s)} \right| \leq \gamma \quad (3.18)$$

where  $L_N(s) = P_N(s)K_N(s)$  and  $\gamma$  is a constant. Here  $P_N(s)$  denotes the transfer function of the plant and  $K_N(s)$  the transfer function of the feedback controller. The subscript  $N$  is used to avoid confusion with previous sections, and emphasises that the discussion here is based on the Nichols chart. Let the nominal plant for this system be given by:

$$P_{N_o}(s) = \frac{N_o(s)\hat{Z}_o(-s)}{D_o(s)\hat{P}_o(-s)} \quad (3.19)$$

In order to achieve a stable/minimum phase system, without affecting the magnitude frequency response of  $P_{N_o}(s)$ , while shifting its phase response along the horizontal axis, define:

$$A(s) = \frac{\hat{Z}_o(-s)\hat{P}_o(s)}{\hat{Z}_o(s)\hat{P}_o(-s)} \quad (3.20)$$

Clearly  $A(s)$  is an all-pass function, i.e.  $|A(j\omega)| = 1$  for all  $\omega \in [0, \infty)$ . Define also:

$$P'_{N_o}(s) = P_{N_o}(s)A^{-1}(s) \quad (3.21)$$

$$= \frac{N_o(s)\hat{Z}_o(s)}{D_o(s)\hat{P}_o(s)} \quad (3.22)$$

and note that the system in equation 3.22 is stable and minimum-phase. Now,

$$L_N(s) = P_N(s)K_N(s) = P'_{N_o}(s)K_N(s)A(s) = L'_N(s)A(s)$$

where  $L'_N(s) = P'_{N_o}(s)K_N(s)$  and  $P'_{N_o}(s) = P_{N_o}(s)A^{-1}(s)$ . Then, for a general frequency  $\omega$ , the robust stability condition is given by:

$$\left| \frac{L_N(j\omega)}{1 + L_N(j\omega)} \right| = \left| \frac{L'_N(j\omega)}{A(j\omega) + L'_N(j\omega)} \right| \leq \gamma \quad (3.23)$$

Since  $L_N(j\omega) = \frac{P_N(j\omega)}{P_{N_o}(j\omega)}L_{N_o}(j\omega)$ , the robust stability condition in equation 3.23 for a nominal plant is given by:

$$\left| \frac{\frac{P_N(j\omega)}{P_{N_o}(j\omega)}L_{N_o}(j\omega)}{1 + \frac{P_N(j\omega)}{P_{N_o}(j\omega)}L_{N_o}(j\omega)} \right| \leq \gamma \quad (3.24)$$

Substituting  $L_{N_o}(j\omega)$  in terms of  $L'_{N_o}(j\omega)$  we get

$$\left| \frac{\frac{P_N(j\omega)}{P_{N_o}(j\omega)} L'_{N_o}(j\omega) A(j\omega)}{1 + \frac{P_N(j\omega)}{P_{N_o}(j\omega)} L'_{N_o}(j\omega) A(j\omega)} \right| \leq \gamma$$

Substituting  $L_{N_o} = re^{j\theta}$  in the above equation we get

$$\left| \frac{\frac{P_N(j\omega)}{P_{N_o}(j\omega)} A(j\omega_i) r' e^{j(\theta - \arg A(j\omega))}}{1 + \frac{P_N(j\omega)}{P_{N_o}(j\omega)} A(j\omega) r e^{j(\theta - \arg A(j\omega_i))}} \right| \quad (3.25)$$

From equation 3.25

$$\theta' = \theta - \arg A(j\omega) \quad \text{and} \quad r' = r$$

where  $L'_{N_o}(j\omega) = r' e^{j\theta'}$ . It can be noticed from the above equation that the robust bound for the new system  $L'_{N_o}(j\omega)$  can be obtained by shifting the robust bounds for the system  $L_{N_o}(j\omega)$  horizontally by an additional phase of  $-\arg A(j\omega)$ , while the gain remains the same. The new stability line in the Nichols chart after the modification of the system is given by:

$$R_m(\omega) = (\theta, r) : \theta = -(2q + 1)180^\circ + \arg(A^{-1}(j\omega))$$

where  $r > 0$  and  $q = 0, \pm 1, \pm 2, \dots$

*Remark:-* The closed-loop response of the transformed nominal plant  $L'_{N_o}(j\omega)$  is stable if only if the number of net positive and negative crossings of the new stability line  $R_m$  is equal to the difference of the number of right-half-plane zeros and poles of  $L(s)$ . [13]

### 3.3 Uncertainty

The plant model used for design purposes is essentially an approximation of the true plant. Due to this fact the output response of the actual plant varies from that of the modelled plant when the two systems are excited by the same input [46, 62]. In control systems, model uncertainty gives rise to the need for *robust* design, i.e. the stability and performance specifications should be ideally maintained despite the presence of uncertainty in the model.

There are two main classes of uncertainty models. These are (i) Unstructured uncer-

tainty models, and (ii) Structured (parametric) uncertainty models. [38]

In an unstructured uncertainty model it is assumed that the plant's input-output characteristics are known only approximately. Thus this type of uncertainty is typically described by frequency-domain bounds on the magnitude response of the system, e.g. via bounds on the deviation from a nominal response. For a multivariable system, for example, an unstructured uncertainty model implies that we do not know how the uncertainty is distributed among the different entries of the system. Unstructured uncertainty models are typically used to describe high-frequency un-modelled dynamics or loss of phase information at high frequencies. Structured or parametric uncertainty, on the other hand, typically arises due to uncertain parameters in the coefficients of the corresponding transfer function or state-space model of the process.

There are many different ways of modelling unstructured uncertainty in dynamic systems. Two of the most common models are the additive uncertainty model and the multiplicative uncertainty model [38]. Additive uncertainty is represented by:

$$G_t(s) = G(s) + \Delta(s) \quad (3.26)$$

where  $G_t(s)$  is the true plant,  $G(s)$  is the nominal model and  $\Delta(s)$  is the uncertainty. A *Multiplicative* or *Relative uncertainty* model is given in the form:

$$G_t(s) = (I - \Delta(s))G(s) \quad (3.27)$$

where again  $G_t(s)$  is the true model,  $G(s)$  is the nominal model and  $\Delta(s)$  is the uncertainty. In both cases it is assumed that  $\Delta(s)$  is any transfer function satisfying an inequality of the form  $\|\Delta(j\omega)\| \leq \mu(\omega)$  for an appropriate real-valued function  $\mu(\omega)$ . Here  $\|\Delta\|$  represents an appropriate norm of  $\Delta$ , which in general can be assumed to be a matrix. Note that for a multiplicative uncertainty model, uncertainty is defined at the *output* of the plant. Thus this model would be appropriate to represent a system with uncertain or noisy sensors. A dual model of input multiplicative uncertainty could be described by the equation  $G_t(s) = G(s)(I - \Delta(s))$  which could model a system with uncertain actuators. Of course the above distinction is valid only for multivariable systems. In the scalar case where  $G(s)\Delta(s) = \Delta(s)G(s)$  the two models are equivalent. Figures 3.11a and 3.11b show a plant subject to additive and multiplicative uncertainty  $\Delta(s)$ .

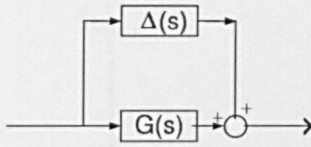


Figure a: Additive Uncertainty

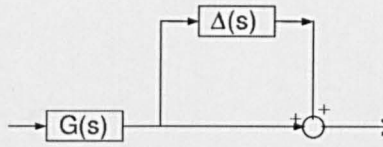


Figure a: Multiplicative Uncertainty

Figure 3.11: *Plant with Additive fig(a) and Multiplicative fig(b) Uncertainty's*

A parametric uncertainty model is typically of the form  $\mathbf{G}(s, \phi)$  and is described by a parameter vector  $\phi$ . The plant's coefficients depend on  $\phi$ , whose elements may take values in certain intervals in the parameter space, e.g.  $\phi_L \leq \phi_i \leq \phi_U$ . As the parameters  $\phi_i$  vary between their lower and upper limits a complete family of systems is generated which describes the permissible structured uncertainty set. A nominal model may be obtained, if desired, by selecting a permissible (fixed) vector  $\phi^0$  such that  $\phi_L \leq \phi_i \leq \phi_U$  for all  $i$ ,  $\mathbf{G}(s, \phi^0)$ .

### 3.3.1 Parametric Uncertainty

To design a robust control system we first have to define the model's region of uncertainty. This is a set which includes the actual plant and all neighboring admissible plants. The uncertainty can be described in parametric space. For example, consider a simple second order system,

$$\mathcal{G}(s) = \frac{ak}{s(s+a)} \quad (3.28)$$

where  $k$ , and  $a$  are assumed to be real uncertain parameters which vary independently in the intervals  $1 \leq k \leq 10$  and  $1 \leq a \leq 10$  respectively (*Note:* This example will be developed in Chapter 4 of the thesis to illustrate the general QFT design approach). The uncertainty is represented by the shaded rectangle in figure 3.12. The variation in parameter  $k$  and  $a$  implies that both the gain and phase of the plant are uncertain at each frequency.



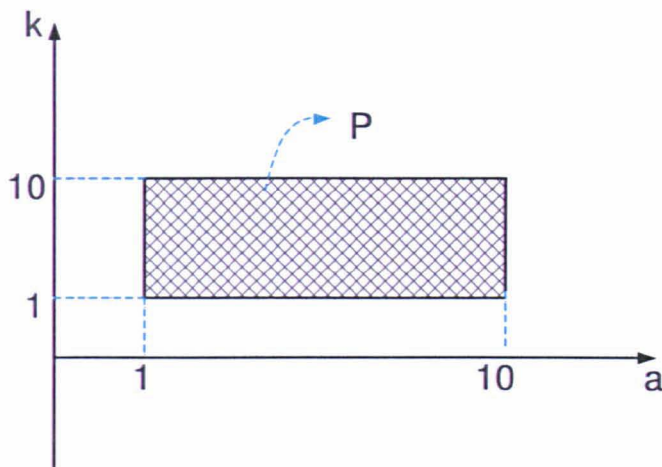


Figure 3.12: *Parameter Uncertainty*

### 3.3.2 Frequency Domain

The uncertainty associated with a plant can also be represented in the frequency domain either in the Nyquist diagram or the Nichols chart. The next section describes the plant uncertainty representation in the Nyquist plane.

#### Nyquist Plot

Figure 3.13 shows a typical plot of uncertainty in the frequency domain in the Nyquist plane, together with the nominal frequency response  $G_o(j\omega)$ .

The uncertain region at a given frequency  $\omega_i$  is represented by a circle of radius  $r_i$  around the nominal plant  $G_o(j\omega_i)$ . In this case this represents additive uncertainty of the form  $G(j\omega_i) = G_o(j\omega_i) + \Delta(j\omega_i)$  with  $|\Delta(j\omega_i)| \leq r_i$ . The uncertainty region may not necessarily be a circle, for example if phase information is available for  $\Delta(j\omega_i)$ .

Provided the nominal closed-loop system is stable, the closed-loop system corresponding to any perturbed plant will also be stable so long as the number of encirclements of the  $-1$  point by the entire uncertainty envelope does not change, since the Nyquist stability criterion will also be satisfied by every perturbed plant. Thus a sufficient

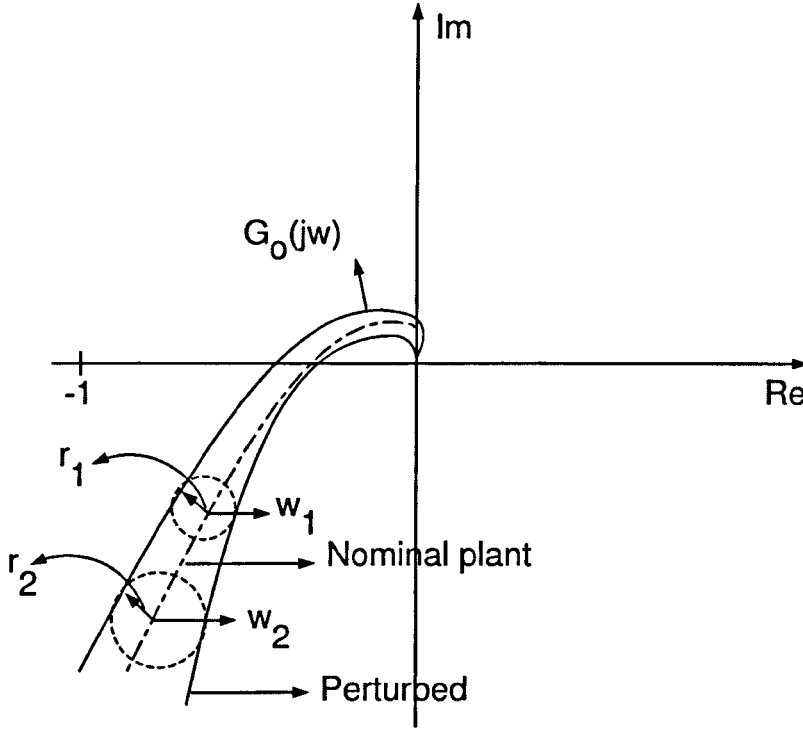


Figure 3.13: *Uncertainty Represented on Nyquist plot*

condition for robust stability is

$$|1 + K(j\omega)G(j\omega)| \neq 0$$

which is assumed to hold for every uncertain plant  $G$  and every frequency  $\omega$ .

A relation between robust stability and the closed-loop response of the plant can be derived based on this condition [23]. For the system shown in figure 3.11 the stability condition can be written as:

$$|1 + K(j\omega)G_o(j\omega)(1 + \varepsilon\Delta)| \neq 0 \quad (3.29)$$

for every  $\varepsilon$  with  $0 \leq \varepsilon \leq 1$  and every  $\omega$  with  $0 \leq \omega \leq \infty$ . We can interpret this condition as follows: As  $\varepsilon$  increases continuously between 0 and 1, the “fuzzy” region representing the frequency response of the uncertain system deforms continuously without ever crossing the  $-1$  point. Since the nominal closed-loop system ( corresponding to  $\varepsilon = 0$ ) is assumed stable, and the number of encirclements around the critical point do not change, the perturbed system will also be stable. Equation 3.29 is true if and only if:

$$|1 + K(j\omega)G_o(j\omega)(1 + \varepsilon\Delta(j\omega))| > 0$$



from which we get

$$|K(j\omega)G_o(j\omega)[(K(j\omega)G_o(j\omega))^{-1} + 1 + \varepsilon\Delta(j\omega)]| > 0$$

The last condition holds, if  $|K(j\omega)G_o(j\omega)| \neq 0$  and

$$|(K(j\omega)G_o(j\omega))^{-1} + 1 + \varepsilon\Delta(j\omega)| > 0$$

which gives

$$|1 + (K(j\omega)G_o(j\omega))^{-1}| > |\Delta(j\omega)| \quad (3.30)$$

Equation 3.30 gives a sufficient condition for robust stability. It can be shown [61] that this is also a necessary condition in the sense that a permissible  $\Delta$  can be found on the boundary of the uncertainty set which de-stabilises the system. This is subject to the assumption that  $G_o$  and  $G_o(1 + \Delta)$  have the same number of right-half-plane poles [43].

This chapter has shown in detail the translation of the established Nyquist stability criterion to a stability criterion in the Nichols chart.

# Chapter 4

## QFT Design

In this chapter the QFT design procedure is described in detail. The first section of this chapter is used to define the system *design specifications*. These are given in the form of upper and lower bounds on the target closed-loop frequency responses, normally specified with the help of a prototype second order transfer function. These specifications include tracking design objectives (in the form of upper and lower bounds in the magnitude frequency response of the complementary sensitivity function which determine the allowable gain spread across all frequencies), disturbance rejection (upper bound on the sensitivity function), etc. Having defined the design specifications, the uncertainty of the system is mapped in the frequency domain (Nichols chart) in the form of *Uncertainty Templates*. These are contours in the Nichols chart which specify the magnitude and phase spread of the plant at a number of discrete frequencies. At this stage we also choose a nominal plant, corresponding to a specific parameter vector within the admissible range. In theory, since the design specifications must be satisfied for every plant in the uncertainty set, the choice of nominal plant can be made arbitrarily.

Next the robust stability and performance bounds are defined. This involves translating the specifications in an open-loop setting, i.e. specifying regions in the Nichols chart in which the nominal open-loop frequency response must lie to meet the predefined specifications (formulated in terms of the target closed-loop transfer function). The robust stability specifications can be translated into a universal High Frequency Region, called the  $\mathcal{U}$ -Contour. This is a closed-contour in the Nichols chart containing the critical point. The robust performance specifications (tracking and disturbance rejection) are mapped into *Tracking Bounds* and *Disturbance-rejection Bounds*. These

are typically open contours and divide the chart into two regions, the low-gain and the high-gain region. They specify the minimum amount of nominal open-loop gain (for each possible phase) which is required to achieve the corresponding objective (tracking or disturbance rejection) at each design frequency. Taking the point-wise maximum of each pair of contours for each design frequency, defines the overall bounds of the QFT problem ( "*Horowitz templates*").

The design proceeds via a *Loop-Shaping* procedure to obtain the desired feedback controller. The open-loop frequency response must now lie on or above the corresponding Horowitz template at each design frequency, it must not penetrate the  $U$ -contour and must satisfy the Nyquist stability criterion in terms of the number and direction of encirclements around the critical point. In order not to over-design the system, the open-loop frequency response at the design frequencies should be as close as possible to the Horowitz templates and the  $U$ -contour. Finally, the design is concluded by designing the *Pre-Filter*.

QFT is a design methodology which describes the effects of feedback quantitatively. The objective is to design a robustly - stable system whose output variation is kept within acceptable limits in the presence of plant uncertainty and external disturbances. It is assumed that plant  $P$  is associated with a certain amount of uncertainty, described by the set  $P \in \mathcal{P}$ .

The space  $\mathcal{P}$  may denote either parametric or unstructured uncertainty. In addition, external disturbances are assumed to be represented by the set  $D \in \mathcal{D}$ . The objective of the design is to guarantee that the input - output transfer ratio  $T_R(s)$  and output - disturbance ratio  $T_D(s)$  are members of appropriate sets  $\mathcal{T}_R$  and  $\mathcal{T}_D$ , where  $\mathcal{T}_R$  is the acceptable set of control ratios and  $\mathcal{T}_D$  is the acceptable set of disturbance ratios, for all plants  $P \in \mathcal{P}$  and disturbances  $D \in \mathcal{D}$ . Here we have defined:

$$\mathbf{T}_R(s) = \frac{Y(s)}{R(s)} \quad \text{and} \quad \mathbf{T}_D(s) = \frac{Y(s)}{D(s)} \quad (4.1)$$

which denote the input - output and disturbance output ratios defined with reference to figure 1.1. where  $Y(s)$  is the output signal,  $R(s)$  is the reference input and  $D(s)$  is

the disturbance.

## 4.1 Design Specification

Typical design specifications require that the plant response is bounded within predefined frequency bounds, called Tracking bounds, i.e. that the system response is forced to track this model. The tracking models are typically defined via a second order system chosen to satisfy all or just the necessary step response parameters like  $M_p$  (maximum peak),  $t_p$  (peak time),  $t_s$  (settling time),  $t_r$  (rise time) and  $K_M$  (gain) in case of an under-damped system. Design of an over-damped system is based on parameters  $t_s$ ,  $t_r$  and  $K_M$ . A prototype second order system is given by

$$T(s) = K \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where  $\omega_n$  is the natural frequency and  $\zeta$  the damping factor. These parameters determine the transient step response characteristics of the system. Peak time ( $t_p$ ), rise time ( $t_r$ ) and settling time ( $t_s$ ) are related to  $\omega_n$  and  $\zeta$  by the relations:  $t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$ ,  $t_r = \frac{1.8}{\omega_n}$  and  $t_s = \frac{4.6}{\zeta\omega_n}$  for 2% of the response settling time.

Illustration 4.1 The under-damped system is designed with the rise time  $t_r$  of 1 sec and settling time  $t_s$  of 3 sec. By using the above relations we get:

$$T_{RU}(s) = \frac{11.641}{s^2 + 2.66s + 11.641}$$

The over-damped system is designed by fixing the value of the damping coefficient  $\zeta$  to 0.8 and the settling time  $t_s$  to 3 sec, corresponding to the second-order system

$$T_{RL}(s) = \frac{2.75}{s^2 + 2.65s + 2.75}$$

The step response and magnitude frequency response bands are shown in figures 4.1 and 4.2. The response of these two systems taken together from the tracking bound for the design, which means that the acceptable step response  $y(t)$  of the system must ideally lie within the two bounds shown in figure 4.3, although this can not always be achieved in practice. In formulating the QFT design specifications, it is required that the closed-loop magnitude frequency responses (for all uncertain plants) lie within  $|T_{RU}(j\omega)|$  and  $|T_{RL}(j\omega)|$  at every design frequency (frequency of interest), where  $T_{RU}(j\omega)$  is the up-

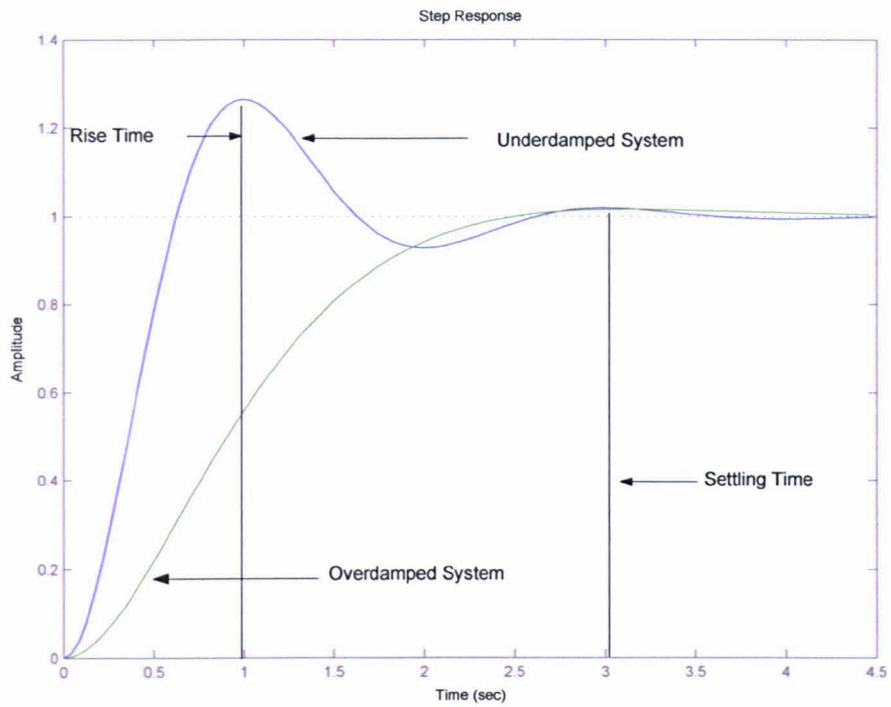


Figure 4.1: *Bounds on step response of the plant*

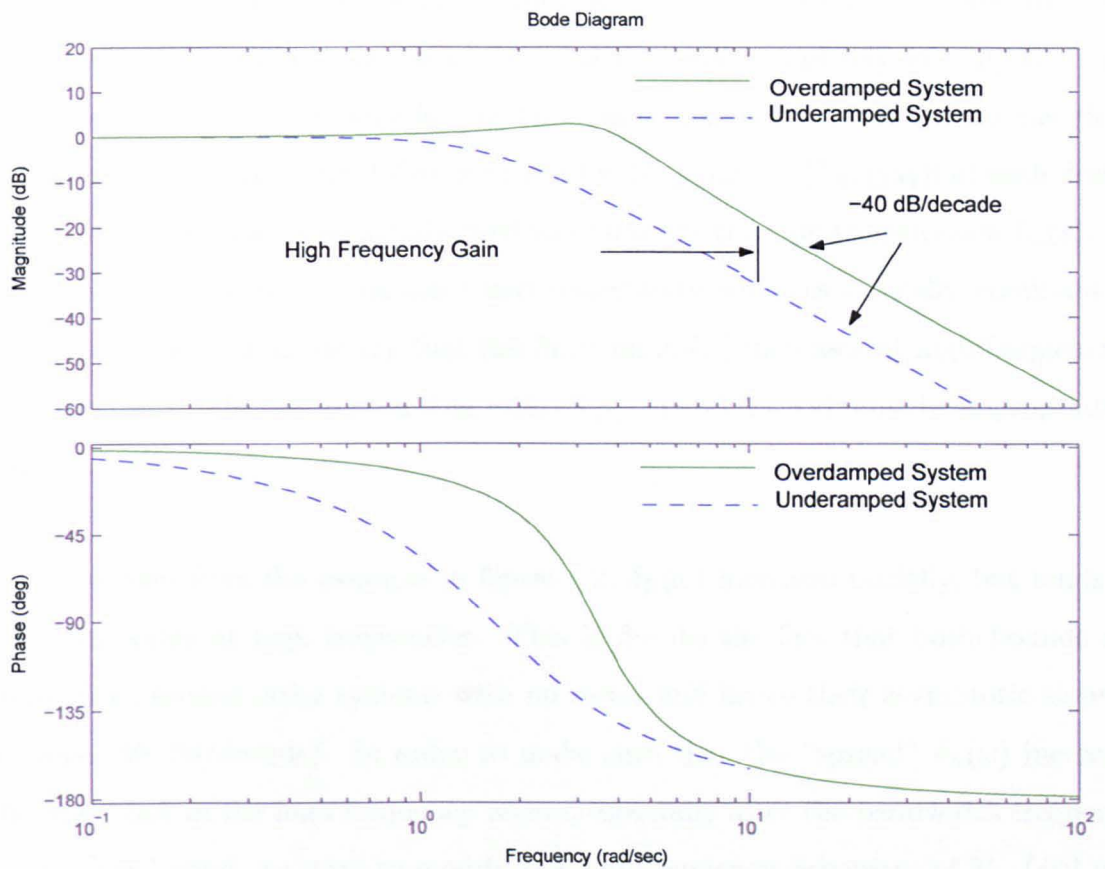


Figure 4.2: *Bounds on the magnitude and phase responses of the plant*

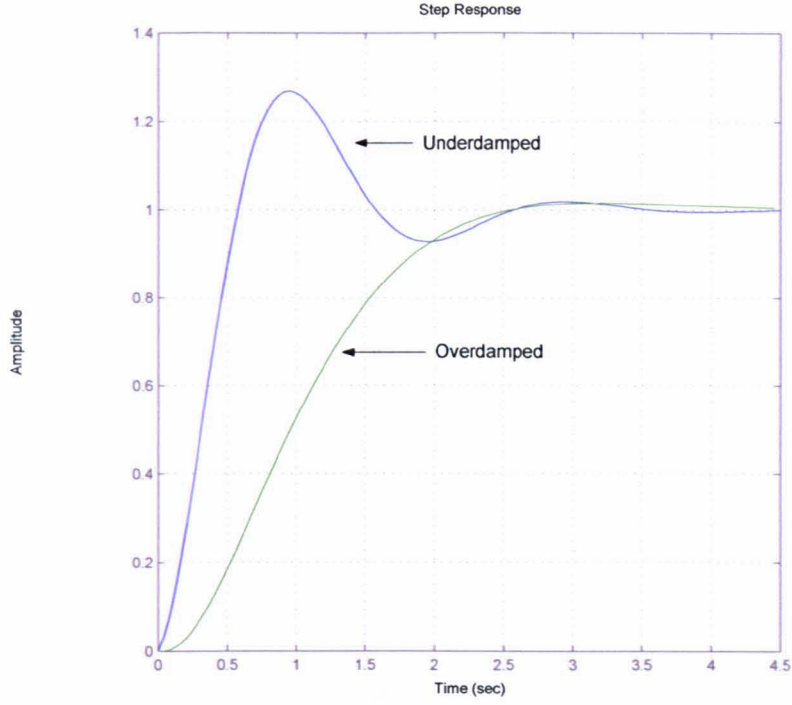


Figure 4.3: *Modified bounds on plant step response*

per bound and  $T_{RL}(j\omega)$  is the lower bound, as shown in figure 4.4. Shaping the step-responses is a more difficult problem which has not been completely solved [52]. Using the (upper and the lower bounds) tracking ratios  $T_{RU}$  and  $T_{RL}$  shown in the Bode plots we next calculate their difference  $\delta_R(\omega_i) = |T_{RU}(j\omega_i)| - |T_{RL}(j\omega_i)|$  at each design frequencies  $\omega_i$ . This is subsequently used to synthesize the loop transmission  $\mathbf{L}_o(s)$ . To accommodate the effects of unstructured uncertainty which is typically dominant at high frequencies, it is necessary that the function  $\delta_R(\omega)$  increases at high frequencies. To ensure this condition, the tracking ratios  $\mathbf{T}_{RU}(s)$  and  $\mathbf{T}_{RL}(s)$  must be appropriately modified.

As can be seen from the example in figure 4.2,  $\delta_R(\omega)$  increases initially, but tends to a constant value at high frequencies. This is due to the fact that both bounds are derived from second order systems with no zeros, and hence their asymptotic slope is the same (40 dB/decade). In order to make sure that the “spread”  $\delta_R(\omega)$  increases sufficiently fast in the high frequency region, especially after the bandwidth frequency of the upper bound, we need to modify the high-frequency behaviour of  $T_{RU}(j\omega)$  and  $T_{RL}(j\omega)$  without effecting their initial step-forcing characteristics.

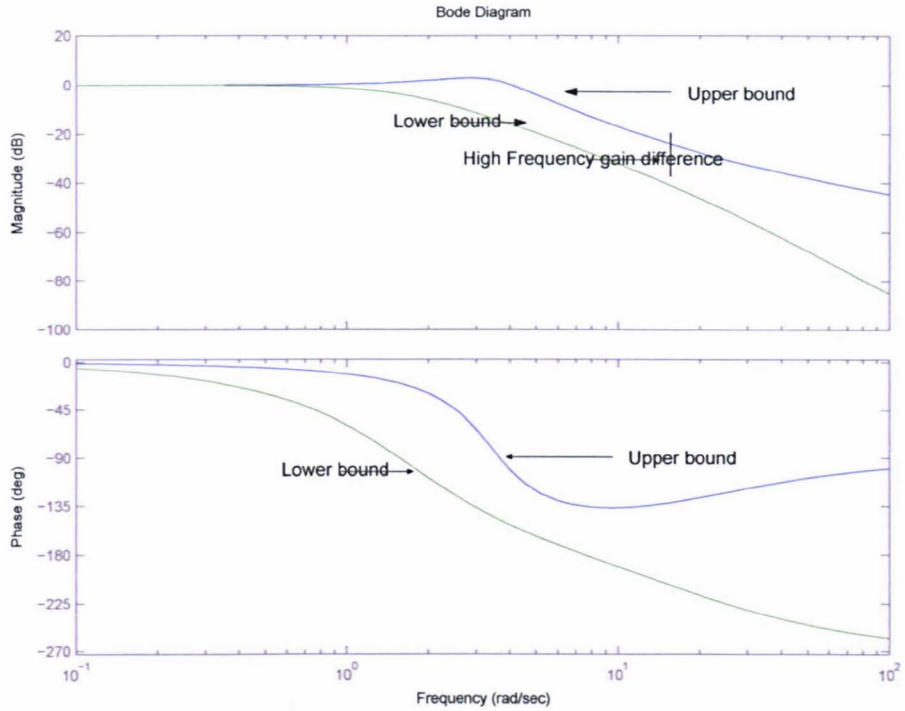


Figure 4.4: *Modified bounds on magnitude and phase responses*

This can be achieved by adding a *fast* stable zero in the case of the upper bound and a *fast* stable pole for the lower bound. The zero and the pole should be placed close enough to the origin, yet, at the same time, they should not significantly affect the step response characteristics of the two systems. These modifications ensure that now we can achieve a sufficient increase in the value of  $\delta_R(j\omega)$  in the high frequency range, as the magnitude Bode plots *diverge* at high frequencies. The new bounds achieved by these modifications are:

$$\begin{aligned} \text{Upper Bound} \quad \mathbf{T}_U(s) &= \frac{0.582s + 11.64}{s^2 + 2.66s + 11.641} \\ \text{Lower Bound} \quad \mathbf{T}_L(s) &= \frac{55}{s^3 + 22.65s^2 + 55.75s + 55} \end{aligned}$$

In the case of the upper bound we have added a stable zero at 20 rads/sec, and for the lower bound a stable pole at 20 rads/sec. It can be seen in figure 4.3 that the step response of the two systems are not significantly affected by these modifications. The Bode plot of the modified systems in figure 4.4 shows that  $\delta_R(\omega)$  has increased in the high frequency range. The adjusted response is now used to form the tracking bounds in the Nichols chart.

## 4.2 Uncertainty Templates

The uncertainty present in the plant will cause a variation in the tracking ratio  $\mathcal{T}_R(s)$  of the system defined in figure 1.1. The tracking ratio of the system is given by:

$$\mathcal{T}_R(s) = \frac{F(s)K(s)G(s)}{1 + K(s)G(s)} = \frac{F(s)L(s)}{1 + L(s)}$$

so that

$$\begin{aligned}\Delta(\log \mathcal{T}_R(s)) &= \log \mathcal{T}_R(s) - \log F(s) \\ &= \log \left( \frac{L(s)}{1 + L(s)} \right)\end{aligned}$$

The main aim of the procedure is to reduce the variation in the tracking ratio due to the uncertainty in the plant, thus restricting it in the region between the tracking bounds defined previously.

The uncertainty of the plant must be represented in the frequency domain, so that the design can be carried out using the Nichols chart. This is done by taking samples at various points on the plot  $\mathbf{G}$  in figure 4.5, to obtain the gain  $\log |\mathbf{G}(j\omega_i)|$  dB and the phase  $\arg \mathbf{G}(j\omega_i)$  for different frequencies  $\omega_i$ . The choice of the frequencies  $\omega_i$  is based on the bandwidth of the system. The plant with nominal loop parameters is denoted as  $\mathbf{G}_o$ .

The nominal loop transmission for the plant  $\mathbf{G}(j\omega_i)$  is given by

$$\log |\mathbf{L}_o| = \log \mathbf{K}\mathbf{G}_o = \log \mathbf{K} + \log \mathbf{G}_o \quad (4.2)$$

Thus the variation  $\delta_p(j\omega_i)$  in  $\log \mathbf{L}(j\omega_i)$  at  $\omega = \omega_i$  is given by

$$\delta_p(j\omega_i) = \log \mathbf{L}(j\omega_i) - \log \mathbf{L}_o(j\omega_i) = \log \mathbf{G}(j\omega_i) - \log \mathbf{G}_o(j\omega_i) \quad (4.3)$$

since the controller is assumed to have zero uncertainty. The corresponding phase variation is:

$$\Delta \arg \mathbf{L}(j\omega_i) = \arg \mathbf{L} - \arg \mathbf{L}_o = (\arg \mathbf{K} + \arg \mathbf{G}) - (\arg \mathbf{K} + \arg \mathbf{G}_o) = \arg \mathbf{G} - \arg \mathbf{G}_o \quad (4.4)$$



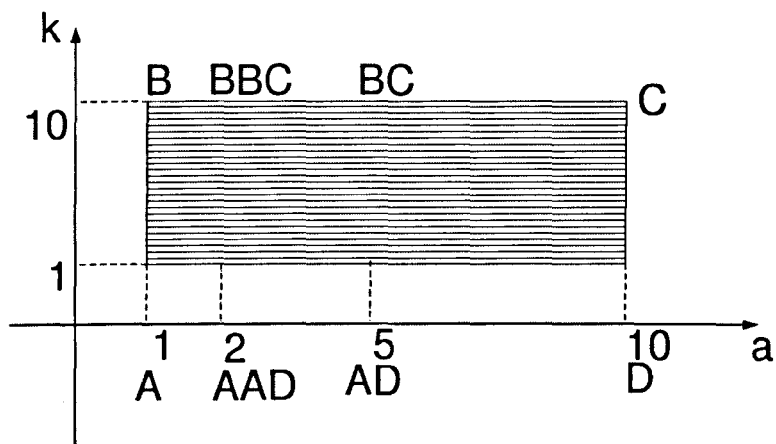


Figure 4.5: *Parameter Uncertainty*

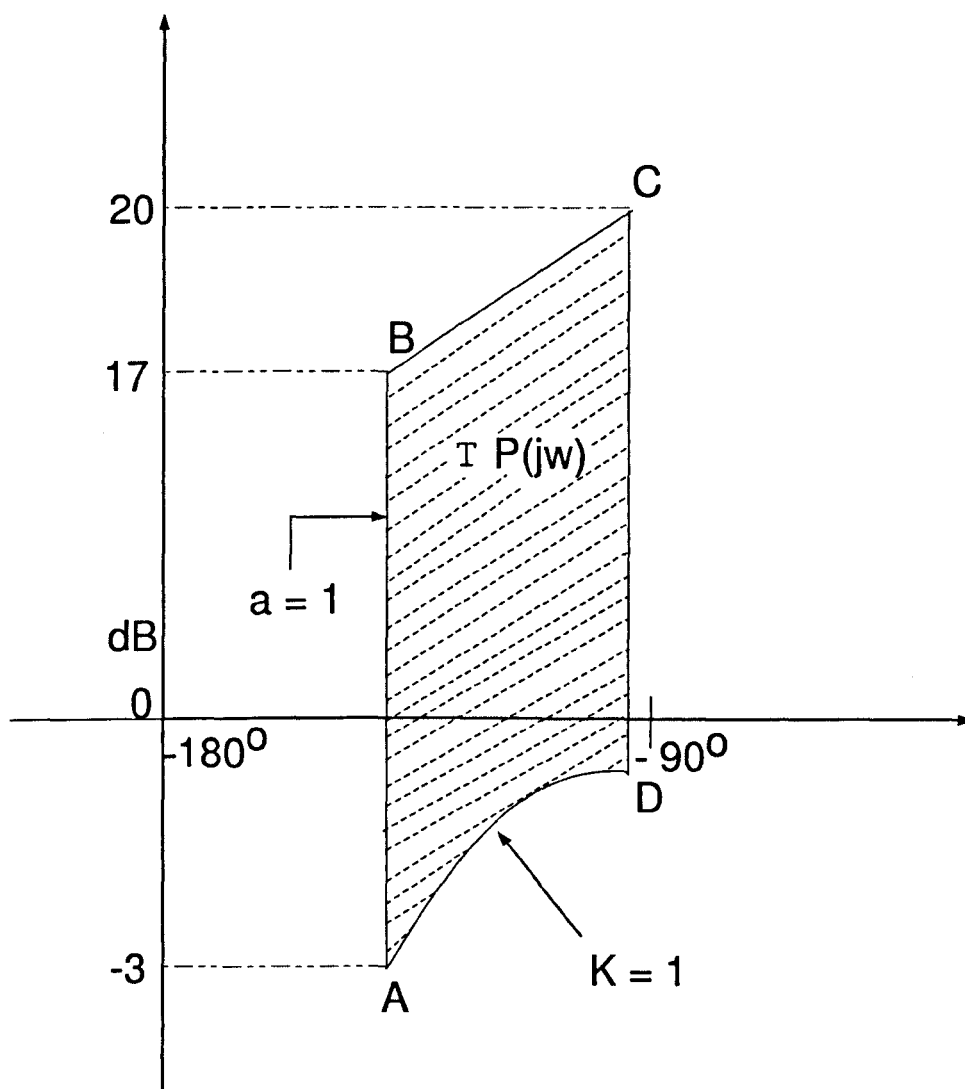


Figure 4.6: *Uncertainty Template*

As an example consider the second order plant:

$$G(s) = \frac{a + s}{(s + b)(s + c)}$$

The magnitude  $|G(j\omega)|$  is given by

$$|G(j\omega)| = \frac{\sqrt{\omega^2 + a^2}}{\sqrt{\omega^2 + b^2}\sqrt{\omega^2 + c^2}}$$

and its phase is

$$\arg G(j\omega) = \tan^{-1} \left( \frac{\omega}{a} \right) - \tan^{-1} \left( \frac{\omega}{b} \right) - \tan^{-1} \left( \frac{\omega}{c} \right)$$

where  $\omega$  is an arbitrary frequency.

*Illustration 4.2* In order to construct the Horowitz templates we chose eight frequency points  $\omega \in \{0.5, 1, 2, 3, 5, 10, 30, 60\}$  rads/sec such that they cover the operating region of the system; the corresponding magnitudes and phases are shown in Table 1.

$(\omega_i)_{Point}$	0.5	1	2	3	5	10	30	60
$(\log P)_A$	5	-3	-13	-19.54	-28.1	-40	-59.08	-71.12
$(\arg P)_A$	-116.6	-135	-153.5	-161.56	-168.7	-174.3	-178.09	-179.04
$(\log P)_B$	25	17	7	0.45	-8.01	-20	-39.08	-51.12
$(\arg P)_B$	-116.6	-135	-153.5	-161.56	-168.7	-174.3	-178.09	-179.04
$(\log P)_{BBC}$	25.8	19	11	5.33	-2.6	-14.2	-33.06	-45.1
$(\arg P)_{BBC}$	-104.1	-116.6	-13.5	-146.3	-15.2	-168.7	-176.18	-178
$(\log P)_{BC}$	26	19.8	13.3	9.12	3	-7	-25.22	-37.176
$(\arg P)_{BC}$	-95.7	-101.3	-11.8	-120.96	-135	-153.5	-170.5	-175.236
$(\log P)_C$	26	20	13.8	10.08	5	-3	-19.54	-31.24
$(\arg P)_C$	-93	-95.7	-101.3	-106.69	-116.6	-135	-161.56	-170.5
$(\log P)_D$	6	-0.04	-6.2	-9.91	-14.95	-23	-39.54	-51.24
$(\arg P)_D$	-93	-95.7	-101.3	-106.69	-116.6	-135	-161.56	-170.5
$(\log P)_{AD}$	6	-0.17	-6.67	-10.87	-17	-27	-45.22	-57.176
$(\arg P)_{AD}$	-95.7	-101.3	-111.8	-120.96	-135	-153.5	-170.5	-175.23
$(\log P)_{ADD}$	5.65	0.97	-9	-14.66	-22.6	-34.2	-53.08	-65.110
$(\arg P)_{ADD}$	-104.1	-116.6	-135	-146.3	-158.2	-168.7	-176.18	-178

Table 4.1: Magnitude and phase values for corresponding frequencies

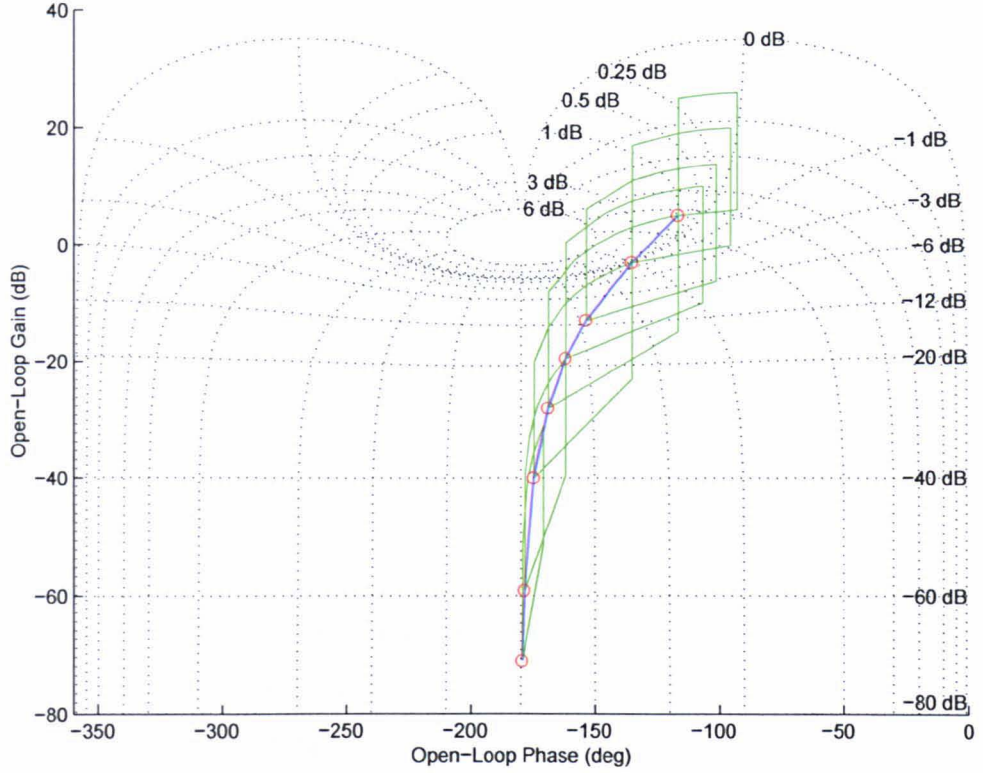


Figure 4.7: *Nominal Open-loop and Uncertainty Templates*

The translation of the uncertainty from the parameter space to the Nichols chart is shown in Figure 4.5 and 4.6.

The magnitude and phase values shown in table 4.1 can be plotted as templates in the Nichols chart for the eight selected frequencies, thus resulting in eight uncertainty templates which will be used in the design.

Apart from these templates we need to specify the nominal plant  $\mathbf{G}_o(j\omega)$ , which in our example corresponds to the transfer function obtained by substituting  $a = 1$  and  $k = 1$  in the equation 3.28 for uncertain parameters. Figure 4.7 shows the nominal frequency response of the plant on Nichols chart. The points indicated by circles correspond to the eight chosen frequencies.

### 4.2.1 U-Contour

Design specifications for the QFT design procedure are normally defined in terms of tracking bounds and disturbance rejection bounds at a finite set of frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$ . However, we still need to maintain a *minimum amount* of damping at high frequencies  $\omega \geq \omega_N$ . This minimum damping corresponds to a resonance peak in the closed-loop magnitude frequency response, or equivalently, to a maximum  $M$ -value on the Nichols chart. To enforce this requirement, we have to ensure that the high-frequency response of the open-loop system does not penetrate the  $M$ -circle of an appropriate value. However the open-loop system itself is uncertain (due to the uncertainty of the plant). When the uncertainty is *parametric*, the high frequency templates typically tend to a vertical line on the Nichols chart (gain variation but fixed phase). Therefore, to ensure a minimum worst-case damping (i.e. a minimum amount of damping for every uncertain plant), we must ensure that the high-frequency part of the nominal open-loop frequency response does not penetrate the contour which is obtained by translating the lower part of the  $M$ -circle downwards by an appropriate amount, specified by the high frequency uncertainty templates. This region is called the *Universal High Frequency Bound* (UHFB) [58].

We enforce the minimum damping condition via an  $M$ -circle with  $M > 1$ . Since in this case the  $M$ -circle does not contain the origin in the Nyquist plane, in the Nichols chart it is a closed contour defined only for an interval of phases, and is symmetric about the phase line  $\phi = -180^\circ$ .

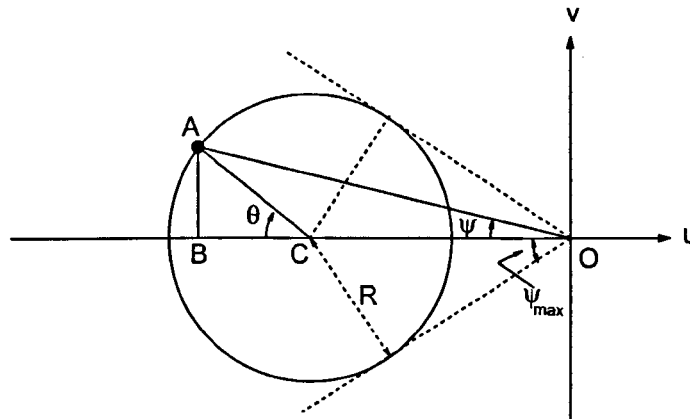


Figure 4.8:  $M$ -circle in Nyquist plane ( $M > 1$ )

By drawing the tangents to the circle from the origin (see figure 4.8), it is clear that

$$\psi_{\max} = \sin^{-1} \left( \frac{1}{M} \right)$$

and hence the  $M$ -circle is defined on the Nichol's chart only for the phase interval

$$-180^\circ - \sin^{-1} \left( \frac{1}{M} \right) \leq \phi \leq -180^\circ + \sin^{-1} \left( \frac{1}{M} \right)$$

Next we need to derive the equation of the  $M$ -circle in terms of magnitude ( $m$ ) and phase ( $\phi$ ), where

$$m = \sqrt{u^2 + v^2} \quad \text{and} \quad \phi = \arctan \frac{v}{u}$$

in which  $u$  and  $v$  denote the real and imaginary variables, respectively. Referring to Figure 4.8, it follows by simple geometry that

$$\begin{aligned} m \cos \psi &= \frac{M^2}{M^2 - 1} + \frac{M}{M^2 - 1} \cos \theta \\ m \sin \psi &= \frac{M}{M^2 - 1} \sin \theta \end{aligned}$$

Eliminating variable  $\theta$  using the trigonometric identity  $\sin^2 \theta + \cos^2 \theta = 1$  results in the second-order equation:

$$m^2 - \frac{2M^2 \cos \psi}{M^2 - 1} m + \frac{M^2}{M^2 - 1} = 0$$

which can be solved as:

$$m = \frac{M^2}{M^2 - 1} \left( \cos \psi \pm \sqrt{\frac{1}{M^2} - \sin^2 \psi} \right)$$

Thus, using the substitution  $\phi = -180^\circ - \psi$ , the  $M$  circle in the Nichols chart is a closed contour which may be decomposed into the union of the graphs of the two functions,

$$M^+(\phi) = 20 \log_{10} \left( -\cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right) + 20 \log_{10} \left( \frac{M^2}{M^2 - 1} \right)$$

and

$$M^-(\phi) = 20 \log_{10} \left( -\cos \phi - \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right) + 20 \log_{10} \left( \frac{M^2}{M^2 - 1} \right)$$

Following Horowitz [29], the  $\mathcal{U}$  contour is obtained by translating  $M^-(\phi)$  vertically by  $V_\infty$  dB's, where

$$V_\infty = \lim_{\omega \rightarrow \infty} \left( \max_{p \in \mathcal{P}} |G(p, j\omega)|_{\text{dB}} - |G(p_o, j\omega)|_{\text{dB}} \right) \quad (4.5)$$

This gives the  $\mathcal{U}$  contour as the union of the graphs of the two functions:

$$\mathcal{U}^+(\phi) = M^+(\phi) \quad \text{and} \quad \mathcal{U}^-(\phi) = M^-(\phi) - V_\infty$$

over the phase interval

$$\phi_l := -180^\circ - \sin^{-1}\left(\frac{1}{M}\right) \leq \phi \leq -180^\circ + \sin^{-1}\left(\frac{1}{M}\right) := \phi_h$$

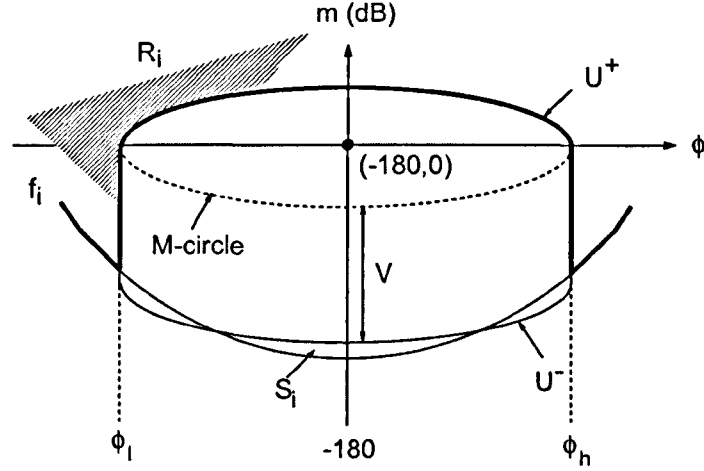


Figure 4.9:  $M$ -circle and  $\mathcal{U}$ -contour in Nichols chart ( $M > 1$ )

*Illustration 4.3* In illustration 4.2, suppose we choose the nominal plant to correspond to parameter values  $k = a = 1$ . By substituting these values into equation 4.5,  $V_\infty$  can be calculated as:

$$\begin{aligned} V_\infty &= 20 \log(ka)_{\max} - 20 \log |j\omega|^2 - 20 \log(ka)_{\min} - 20 \log |j\omega|^2 \\ &= 20 \log(10 \cdot 10) - 20 \log(1 \cdot 1) \end{aligned}$$

and hence  $V_\infty = 40$  dB in this case. Thus, the lower part of the  $M$  circle needs to be extended downwards by this amount as shown in figure 4.10. The  $M$  circle with its extended region defines the  $\mathcal{U}$  contour. During loop-shaping, the nominal open-loop frequency response must lie outside this contour at every frequency. If, in addition, the correct number of encirclements of the critical point is enforced (zero in this example), the design will meet the required robust-stability specifications.

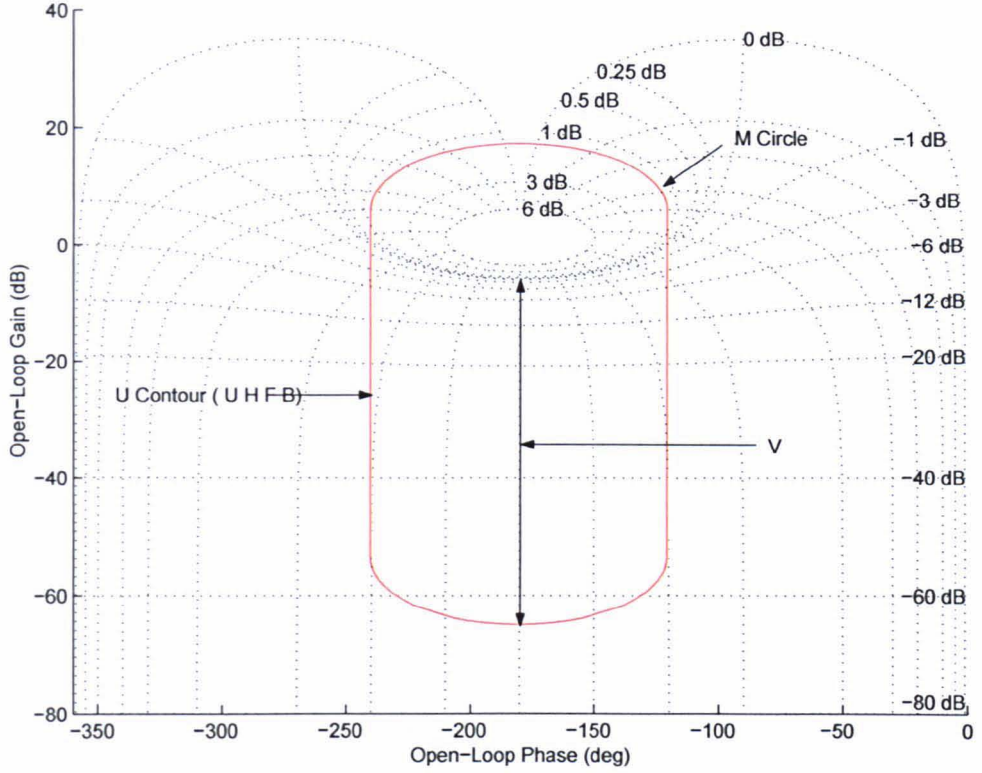


Figure 4.10:  $\mathcal{U}$ -contour

### 4.3 Tracking Bounds

The tracking bounds define the minimum open-loop gain required in order to meet the closed-loop tracking specifications, despite the presence of model uncertainty. Overall,  $n$  bounds are generated, one for each design frequency  $\omega_i$ . The tracking bounds are functions of the open-loop phase and when plotted in the Nichols chart generate  $n$  tracking templates.

In the previous example the uncertain plant was defined as the second order system:

$$G(s; k, a) = \frac{k}{s(s + a)} \quad (4.6)$$

where  $k = [1, 10]$  and  $a = [1, 10]$ . Based on these two uncertain parameters the uncertainty templates  $\mathcal{P}(j\omega_i)$  were produced for a number of frequencies  $\omega_i$ . From figure 4.3 the maximum allowable gain variation of the closed-loop specifications  $\delta_R(\omega_i)$  at a frequency  $\omega_i$  is determined as shown in table 4.2, using the difference

$$\delta_R(\omega_i) = |T_{R_U}(j\omega_i)| - |T_{R_L}(j\omega_i)|$$

between the upper and lower bounds of the closed-loop specifications. The calculation of the tracking bounds is based on  $\delta_R(\omega_i)$  and the plant uncertainty and is summarised next:

**Procedure for calculating tracking bounds:**

- Consider an uncertainty template  $\mathcal{P}$  at a certain design frequency  $\omega_i$  as shown in figure 4.11. Choose a nominal plant  $G_o \in \mathcal{P}$ ; this choice is used throughout the design process.
- Fix the phase of the nominal open-loop system to  $\psi$ , say.
- Move the template vertically in the Nichols chart, along the  $\psi$ -phase line, until:

$$20 \log \max_{p \in \mathcal{P}} \left| \frac{L_o(j\omega)}{1 + L_o(j\omega)} \right| - 20 \log \min_{p \in \mathcal{P}} \left| \frac{L_o(j\omega)}{1 + L_o(j\omega)} \right| = 20 \log \delta_R(\omega_i)$$

This is equivalent to the condition:

$$M_{max} - M_{min} = \delta(\omega_i)$$

where  $M_{max}$  and  $M_{min}$  denote the  $M$  circles of maximum and minimum value, respectively, touched by the uncertainty templates in the Nichol's chart at any given position of the template on the vertical line  $\arg L_o(j\omega_i) = \psi$ . In the pre-CAD era, the required condition (closed-loop gain spread is equal to  $\delta(\omega_i)$ ) required a - rather torturous- procedure, based on the inspection of  $M_{max}$  and  $M_{min}$  from a grid of  $M$ -circles superimposed on the Nichols chart. These days, the step can be implemented via a simple bisection algorithm between appropriate low and upper gain bounds. This algorithm determines the minimum open-loop gain (corresponding to an open-loop phase  $\psi$ ) for which the robust performance specifications are met.

- The template is shifted horizontally (i.e. a different phase  $\psi$  is selected) and the above steps are repeated. The procedure is repeated for every point of a discretised phase grid in the range  $-360^\circ \leq \psi \leq 0^\circ$ .
- The template formed by joining all points determined in this way, when plotted in the Nichol's chart, defines the *tracking bound* of the design for frequency  $\omega_i$ .



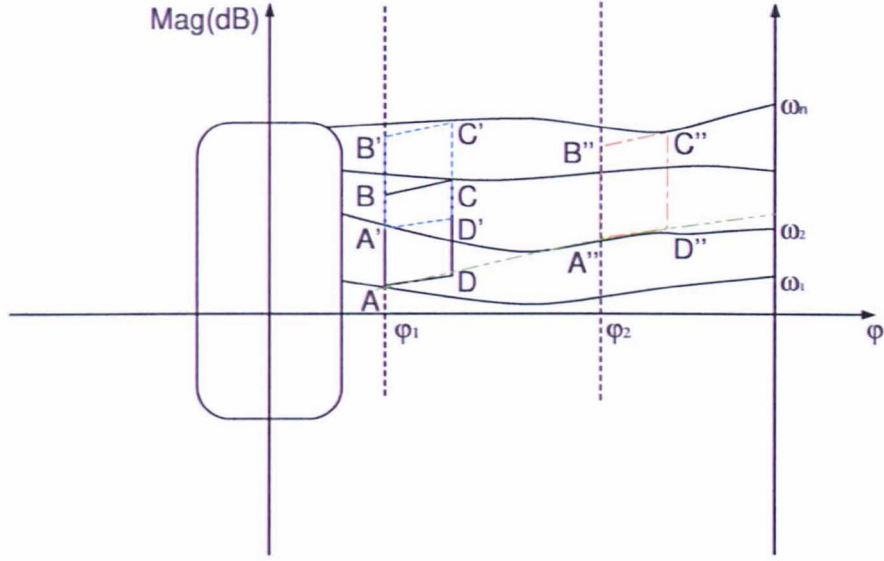


Figure 4.11: *Tracking Bound placements on the Nichols chart using Uncertainty Templates*

- When the above steps are repeated for every design frequency  $\omega_i$ ,  $i = 1, 2, \dots, n$ , the complete set of tracking bounds is obtained.

The horizontal (phase) translation of the template to form the tracking bound is limited only by the phase range in which the UHFB is defined. The nominal open-loop frequency response should not penetrate this region, since the maximum peak of the tracking specification cannot exceed  $M_P$ . Thus the tracking bounds are typically terminated at the boundary of the  $U$  - contour, when these intersect.

$\omega_i \text{ rad/s}$	0.5	1	2	3	5	10	30	60
$\delta(j\omega_i) \text{ dB}$	0.39	1.822	8.02	14.31	15.81	15.4	22.8	33

Table 4.2: Values of  $\delta(j\omega_i)$  over 8 design frequencies

In the loop-shaping phase of the design, the nominal open-loop frequency response is shaped so that it is entirely outside the  $U$ -contour, and its gain at the design frequencies is at least equal to the gain specified by the tracking bounds. Loop-shaping is essentially the function of the feedback controller, which is cascaded with the plant to define the open-loop system. In order to shape the open-loop frequency response, the gain and phase of the controller should be chosen appropriately at the design frequencies. This

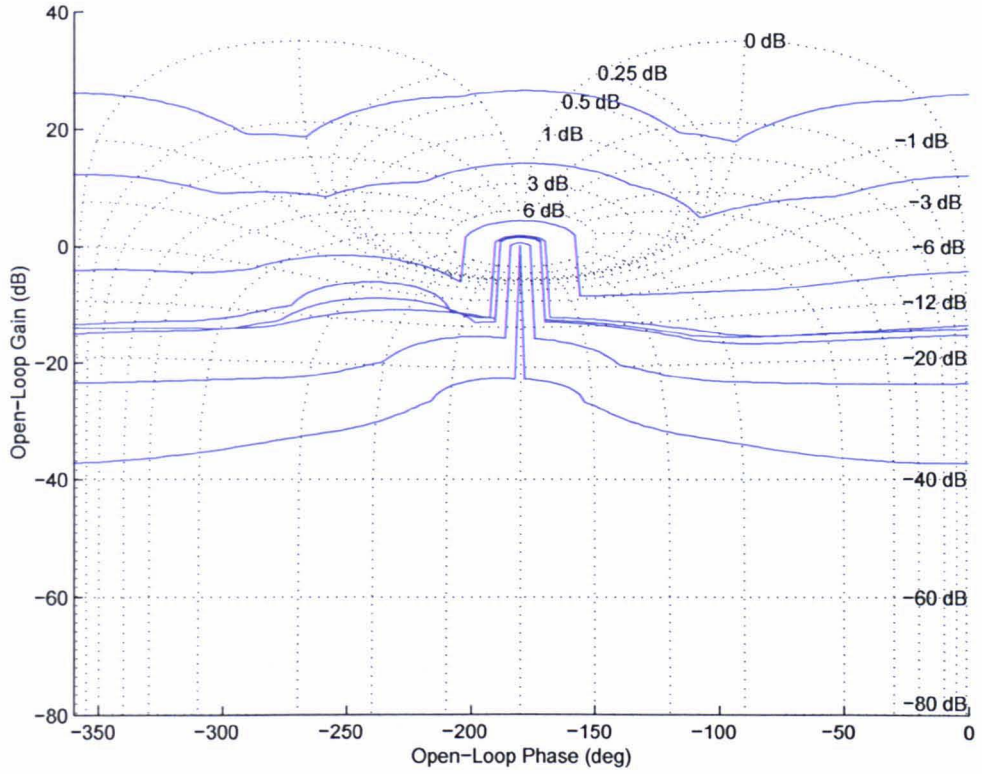


Figure 4.12: *Horowitz Bounds*

may not be a straightforward exercise when the specifications are tight, since the gain and phase variables cannot of course be chosen independently.

*Illustration 4.4* Figure 4.11 shows the process of constructing the bounds. First a nominal plant has to be chosen from the plant uncertainty templates. This choice has to be retained for the rest of the design process. There are no fixed rules for choosing the nominal plant, but by convention the plant corresponding to low gain and maximum phase-lag is normally preferred. First we choose point  $A$  at frequency  $\omega_i = 0.5$  rads/s, where  $20 \log |A(j\omega_i)| = 5$  dB and  $\arg A(j\omega_i) = -116.6^\circ$ . Note that from table 2,  $\delta(j\omega_i) = 0.39$  dB at  $\omega_i = 0.5$ . So the system will tolerate a change of 0.39 dB at frequency  $\omega_i = 0.5$  rads/s. Based on this tolerance value, the uncertainty template  $ABCD$  can be shifted vertically to position  $A'B'C'D'$  such that the maximum difference in the value of  $M$  between any two points of the template is equal to this tolerance level. This process is repeated for various values of phase along the horizontal axis, until sufficient data are obtained in order to construct the bound  $B_R(j\omega_i)$ . If this template intersects the  $U$  contour, we can restrict the phase range of the template, so that it ends at the boundary of the  $U$  contour. The procedure

is repeated for each uncertainty template corresponding to a design frequency, in this case  $\omega_i \in \{0.5, 1, 2, 3, 5, 10, 30, 60\}$  rads/s. The set of templates obtained in this way is displayed in figure 4.12.

## 4.4 Disturbance Rejection

Apart from tracking specifications, a typical design should also reject external disturbances ( $d(s)$  in figure 1.1). Disturbance signals can occur at the input or at the output of the system. Here, only output disturbances are considered; input disturbances can be treated in a similar way. A disturbance can cause the plant to deviate from its operating point, which can make the system unstable. In order to compensate the effect of disturbances, a first or second-order disturbance-rejection model is introduced [17]. The selection of appropriate disturbance rejection models is based on the time-response and frequency-response characteristics of the system. Note that we are interested in robust-disturbance rejection, i.e. the robust-rejection properties of the design should apply for every plant which belongs in the model uncertainty set.

The disturbance-rejection model considered in this section should satisfy the conditions of equation 4.8 for a step disturbance input  $D_o = u_{-1}(t)$ , i.e.

$$|y(t)| \leq \alpha_p \quad \text{for} \quad t \geq t_x \quad (4.7)$$

where  $\alpha_p$  is an appropriate disturbance-rejection level and  $t_x$  is the settling time. Thus, for an initial condition  $y(0) = D_o$  we typically require that:

$$|y(t)| \leq \alpha_p \quad (4.8)$$

for all  $t \geq t_x$ . A typical second-order under-damped disturbance-rejection model in the time domain is of the form:

$$y(t) = D_o e^{-at} \cos(bt)$$

For this model, the disturbance-rejection ratio  $M_D$  can then be defined as:

$$M_D(s) = \frac{Y(s)}{D(s)} = \frac{s(s+a)}{(s+a)^2 + b^2}$$

where

$$a = \frac{1}{t_x} \ln \left( \frac{\alpha_p}{D_o} \right) = \zeta \omega_n$$

$$b = \omega_n (1 - \zeta)^{-\frac{1}{2}}$$

*Illustration 4.5* For the plant model used in this chapter the disturbance model is designed by assuming the following values for the design parameters:

$$D_o = 1$$

$$p_u = 25\%$$

$$\alpha_p = 0.1$$

$$t_x = 2 \text{ sec}$$

Here  $D_o$  is the size of the disturbance step input and  $p_U$  is the peak overshoot or undershoot level of the response to the disturbance signal. The corresponding disturbance model  $M_D(s)$  in this case is:

$$M_D(s) = \frac{s(s + 1.15)}{(s + 1.15)^2 + 2.39^2}$$

The step-response of the disturbance rejection model  $M_D$  is shown in figure 4.13, and its Bode plot in figure 4.14.

## 4.5 Disturbance Bounds $B_D$

In order for the plant to reject robustly the effects of external disturbance, the disturbance rejection model designed in the previous section is transformed to constraints on the (nominal) open-loop frequency response. The corresponding contour in the Nichols chart is known as the disturbance bound  $B_D$ . A procedure similar to the one used for transforming tracking specifications to tracking bounds is used here to transform the disturbance rejection model into disturbance bound: We first select a phase  $\psi$  in the range  $-360^\circ \leq \psi \leq 0^\circ$  for the nominal open-loop system  $L_o(s)$  (note that the nominal plant has already been selected). By sliding the uncertainty template of the plant at design frequency  $\omega_i$  along the vertical phase line  $\psi$ , the minimum open-

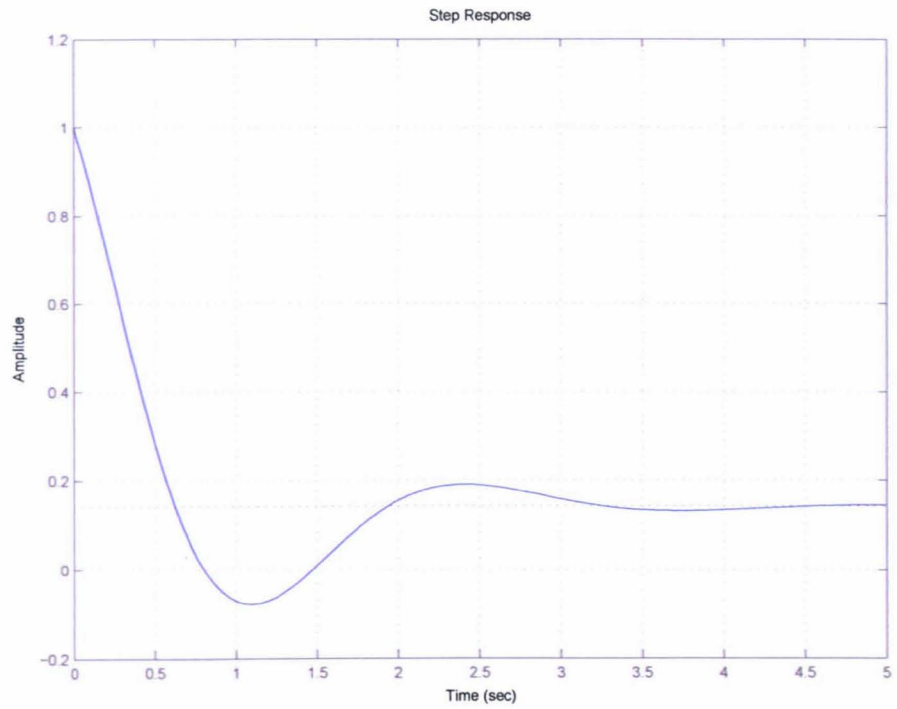


Figure 4.13: *Step response for  $M_d$*

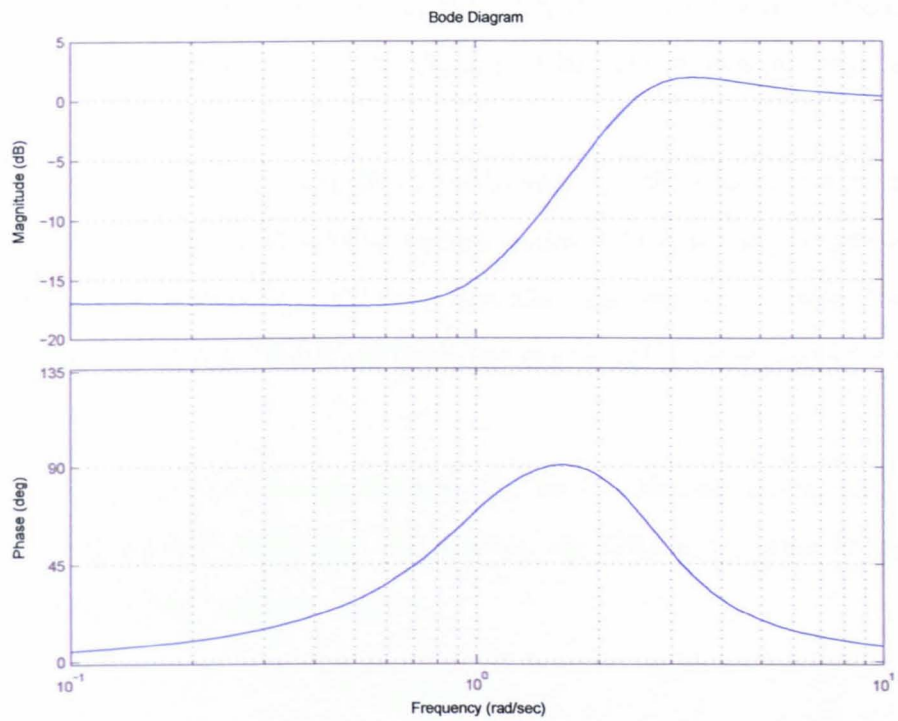


Figure 4.14: *Bode plots for  $M_d$*

loop gain is recorded for which the sensitivity function is “uniformly” sufficiently small, i.e.

$$\left| \frac{1}{1 + K(j\omega_i)G(p, j\omega_i)} \right| \leq |M_D(j\omega_i)|$$

for all  $p \in \mathcal{P}$ , i.e. the minimum gain for which the inequality becomes an equality for some  $p \in \mathcal{P}$ . Again, this step can be performed via a simple bisection algorithm between specified upper and lower gain bounds (and a specified tolerance). Repeating the procedure for each phase  $\psi$  in a discretised grid of the interval  $[-360^\circ \ 0^\circ]$  gives the corresponding disturbance-rejection template for frequency  $\omega_i$ . Repeating for the  $n$  design frequencies gives the whole set of disturbance-rejection bounds.

The disturbance ratio used to obtain the disturbance bounds  $B_D$  is given by:

$$T_D(j\omega) = \frac{1}{1 + L(j\omega)}$$

frequency (rad/sec)	0.5	1.0	2.0	3.0	5.0	10.0	30.0	60.0
Gain $A_i$	20.78	12.5	1.54	2.55	1.59	0.43	0.05	0.01

Table 4.3: Open-loop for robust disturbance rejection

Table 4.3 gives the gain values at the required frequencies for the disturbance rejection model  $M_D$ , which is obtained from the Bode plot for  $M_D$  shown in figure 4.14.

To satisfy both tracking and disturbance-rejection specifications, the nominal open-loop gain needs to lie on or above the overall bound which is the pointwise maximum of the tracking and disturbance-rejection bounds. The overall bounds (for all design frequencies) along with the UHFB will form the overall QFT constraints for the system.

Figure 4.15 shows the disturbance bounds  $B_D$  on the Nichols chart, and figure 4.16 shows the overall bounds which are formed from the UHFB, tracking bounds and disturbance bounds on the Nichols chart.



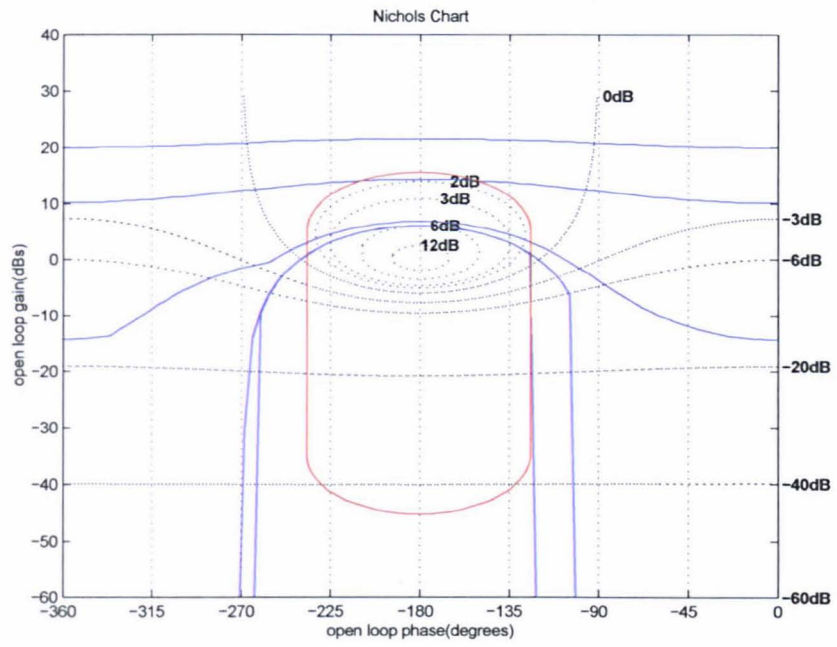


Figure 4.15: *Disturbance bounds  $B_D$*

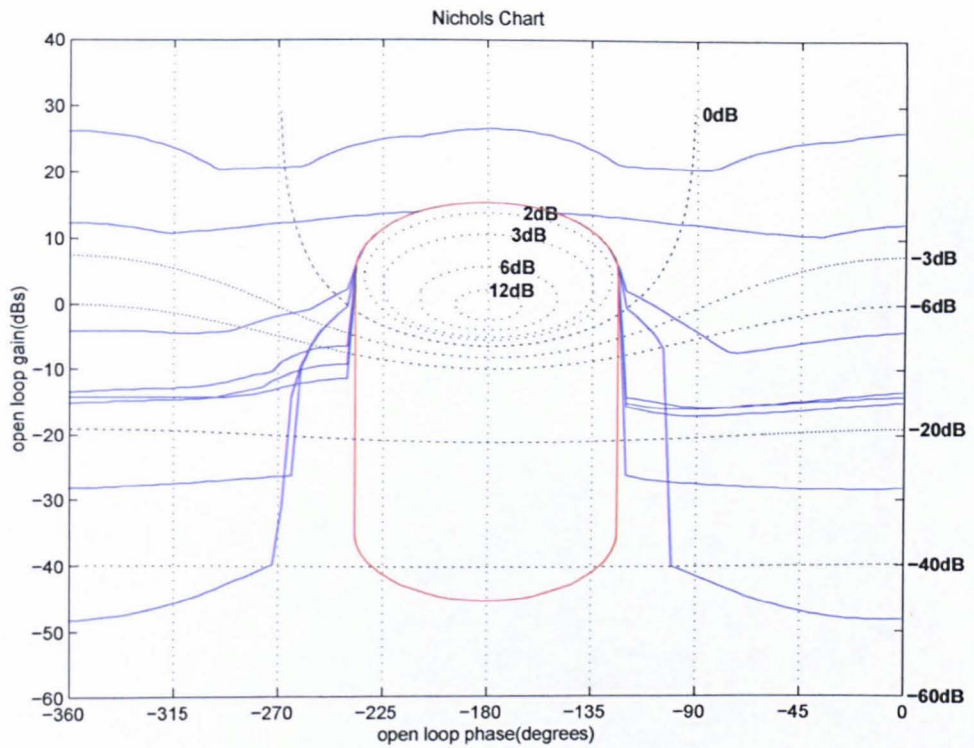


Figure 4.16: *Overall bounds*

## 4.6 Loop-Shaping

Upon obtaining the *overall bounds* the next task is to shape the open-loop frequency response  $L(j\omega)$  so that it satisfies the robust stability and performance requirements.

In this section we review the method of manual loop-shaping. In the next chapter we propose two main algorithms for performing this task automatically. This is particularly useful when there are many design frequencies and the specifications are tight. The automatic approach is also used to induce a measure of *optimality* into the design.

### Typical control structures used in loop-shaping

The types of controllers which can be used for manual loop-shaping are summarised next. These may be used on their own (in simple applications) or in series (cascade) combination to result in more complex controllers.

#### Simple Gain

The simplest type of controller is the simple gain:

$$K(j\omega) = k \quad (4.9)$$

This introduces a simple gain of  $20 \log_{10} |k|$  dB and thus simply shifts the frequency response vertically in the Nichols chart without introducing any phase change (or simply a phase shift of  $-180^\circ$  if  $k$  is negative).

#### Simple Pole or Zero

The frequency response of this controller type is:

$$K(j\omega) = \frac{a}{j\omega + b} \quad (4.10)$$

or,

$$K(j\omega) = b(j\omega + a) \quad (4.11)$$



The use of a first-order controller will result in  $L(j\omega)$  being shifted vertically by a gain of  $-10 \log(\frac{a}{\omega^2 + b^2})$  dB and will introduce a phase shift of  $-\tan^{-1}(\frac{\omega}{b})$  deg (phase lag). Both gain and phase are frequency-dependent in this case. In the case of a zero the additional gain introduced is  $20 \log_{10}(b) + 10 \log_{10}(\omega^2 + a^2)$  dB; the corresponding phase shift is  $\tan^{-1}(\frac{\omega}{a})$  deg. Note that a pole introduces negative phase angle (phase lag), while the zero introduces positive phase (phase lead). Note also that a zero term can be introduced only if the overall controller is proper.

### Lead Lag controller

The more general controller used for loop-shaping is of the phase lead/lag type

$$K(j\omega) = \frac{j\omega + a}{j\omega + b} \quad (4.12)$$

This is a phase lead controller if  $a > b$  and a phase lag controller if  $a < b$ . A phase lead controller introduces positive phase to  $L(j\omega)$  at all frequencies; the maximum phase shift is:

$$\phi = 90^\circ - 2 \arg \tan \left( \sqrt{\frac{a}{b}} \right) \quad (4.13)$$

at frequency

$$\omega = \sqrt{ab} \quad (4.14)$$

A phase lead controller acts as a high pass filter by injecting high gain in the high frequency region. For this reason it can lead to noise amplification and should be used carefully. Conversely, a phase lag controller ( $a < b$ ) introduces negative phase at all frequencies. Its maximum phase lag is also given by  $\phi$  in equation 4.13, attained at frequency  $\omega$  (equation 4.14). In contrast to a phase lead controller, a phase lag controller acts as a low pass filter.

### Second order pole, zero

The third type of controller used for loop-shaping is a second order system with complex poles and no zeros, as shown below. A second order controller will normally not be used on its own, but in cascade with the controllers discussed above. Its frequency response is:

$$K(j\omega) = \frac{1}{\omega_n^2 - \omega^2 + 2j\zeta\omega_n\omega}$$

The second order controller introduces a gain:

$$g(\omega) = -10 \log_{10}((\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2)$$

and phase

$$\phi(\omega) = -\tan^{-1} \left( \frac{2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \right)$$

This type of controller is normally used at high frequencies to introduce a fast “roll-off” to the frequency response of the open-loop system in order to reduce the effects of measurement noise on the output signal or to increase the robust stability margin to multiplicative uncertainty.

The different types of controllers discussed above can be used in any cascade combination to achieve the desired loop-shaping result for  $L(j\omega)$ .

To facilitate the loop-shaping design procedure, a Matlab software tool was developed for designing a cascade of phase lead/lag controllers. For two selected points on the Nichols chart, the required gain and phase shift are determined; these are then used to obtain the required type (lead or lag) and parameters of the controller. The tool also enables the designer to iterate over various design stages, by displaying frequency responses, backtracking on the design by eliminating the last design step, etc. Details can be found in chapter 6.

*Illustration 4.6* In this example we demonstrate the manual loop-shaping procedure. Here we are only interested in achieving the robust stability and performance QFT constraints and not any measure of optimality. From figure 4.10 we can see that in order to achieve the robust stability condition, the open-loop response needs to lie outside the high frequency bound. In order to achieve this, a considerable amount of phase lead (around  $50^\circ$ ) is needed at a gain of 30dB, resulting in the controller:

$$K_1(s) = \frac{s + 11.45}{s + 0.218} \quad (4.15)$$

Although the response has now moved to the right, completely outside the  $U$ -contour, the performance bounds are not yet satisfied, as the response at the eight design frequencies is below the corresponding bounds. This is corrected by increasing the gain by a factor of  $K_2 = 25$ . To further optimise the response further loop-shaping is carried out. The lead/lag controller

$$K_3(s) = \frac{s + 74.86}{s + 85.49} \quad (4.16)$$

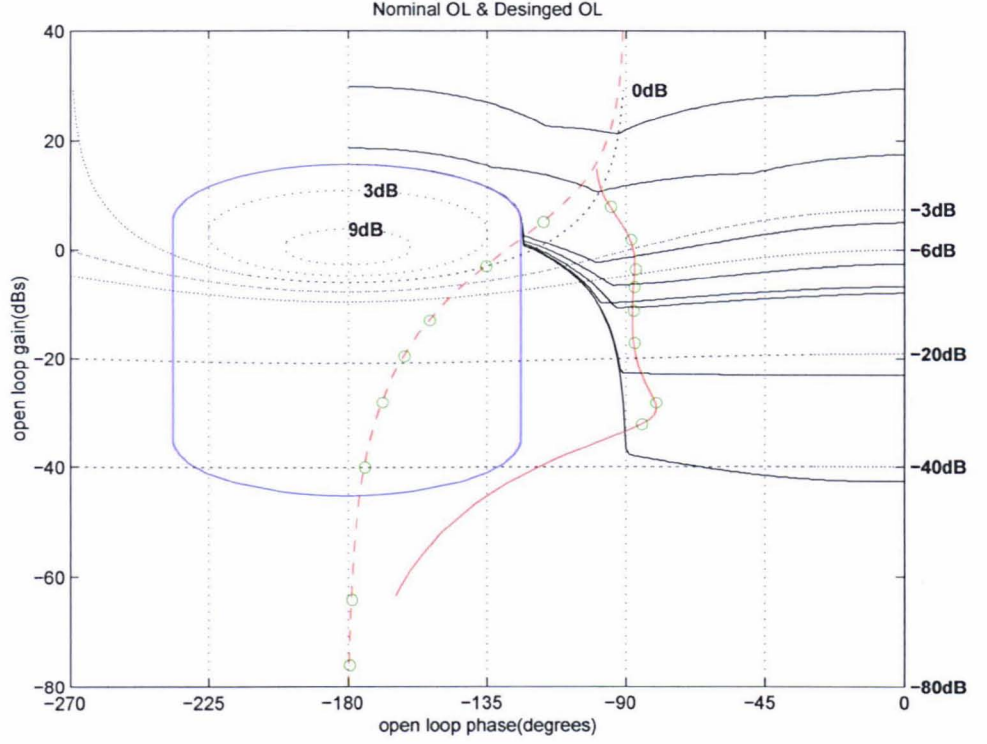


Figure 4.17: *Loop-shaping phase correction*

is used for concluding the loop-shaping procedure. The overall controller is:

$$K(s) = K_1 K_2 K_3(s) = 25 \frac{s + 74.86}{s + 85.49} \frac{s + 11.45}{s + 0.218} \quad (4.17)$$

The initial loop-shaping phase correction plot is shown in figure 4.17. The final open-loop frequency response is shown in figure 4.18 using the designed controller. It can be seen that the shaped response lies to the right of the  $U$ -contour (zero encirclements) and thus achieves robust stability. Moreover, the response at the eight design frequencies lies above the corresponding Horowitz templates, implying that the robust tracking and disturbance-rejection objectives have been met. The proximity of these points to the bounds indicates that the system has not been over-designed (i.e. no excessive gain has been used).

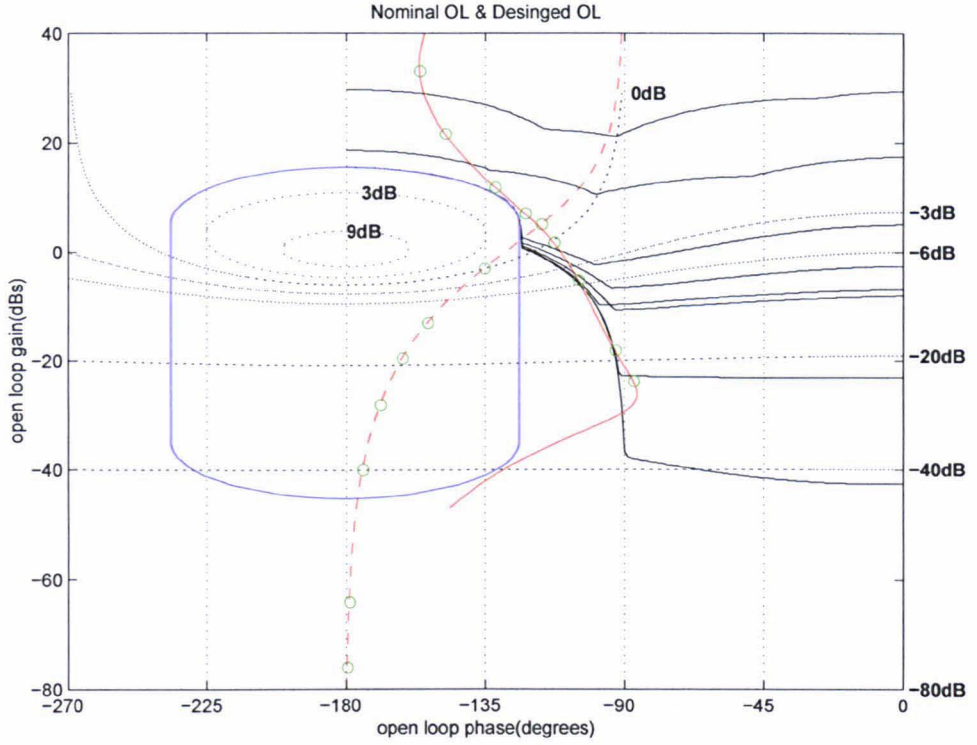


Figure 4.18: *Final Loop-shaping*

## 4.7 Pre-Filter

The principal task of a pre-filter is to scale the responses of the complementary sensitivity function  $\log \left| \frac{PG}{1+PG} \right|$  on the Bode plot, so that they lie within the tracking specifications  $T_{RU}$  and  $T_{RL}$ . Recall that tracking specifications constrain only the gain spread, not the actual gain of the closed-loop system. The required condition can be written as:

$$\max_{p \in \mathcal{P}} \log \left| \frac{PG(j\omega)}{1+L(j\omega)} \right| - \min_{p \in \mathcal{P}} \log \left| \frac{PG(j\omega)}{1+L(j\omega)} \right| \leq T_{RU} - T_{RL} = \delta_{R_i} \quad (4.18)$$

or, equivalently as

$$\alpha(\omega) \leq \left| \frac{PGF(j\omega)}{1+L(j\omega)} \right| \leq \beta(\omega) \quad \forall P \in \mathcal{P} \quad (4.19)$$

for two appropriate functions  $\alpha(\omega)$  and  $\beta(\omega)$ .

The pre-filter transfer function can be obtained by solving a “model-matching” problem over the design frequencies. Its frequency-response must satisfy the inequalities:

$$\log \alpha(\omega) - \min_{p \in \mathcal{P}} \log \left| \frac{PG(j\omega)}{1+L(j\omega)} \right| \leq \log |F(j\omega)| \quad (4.20)$$

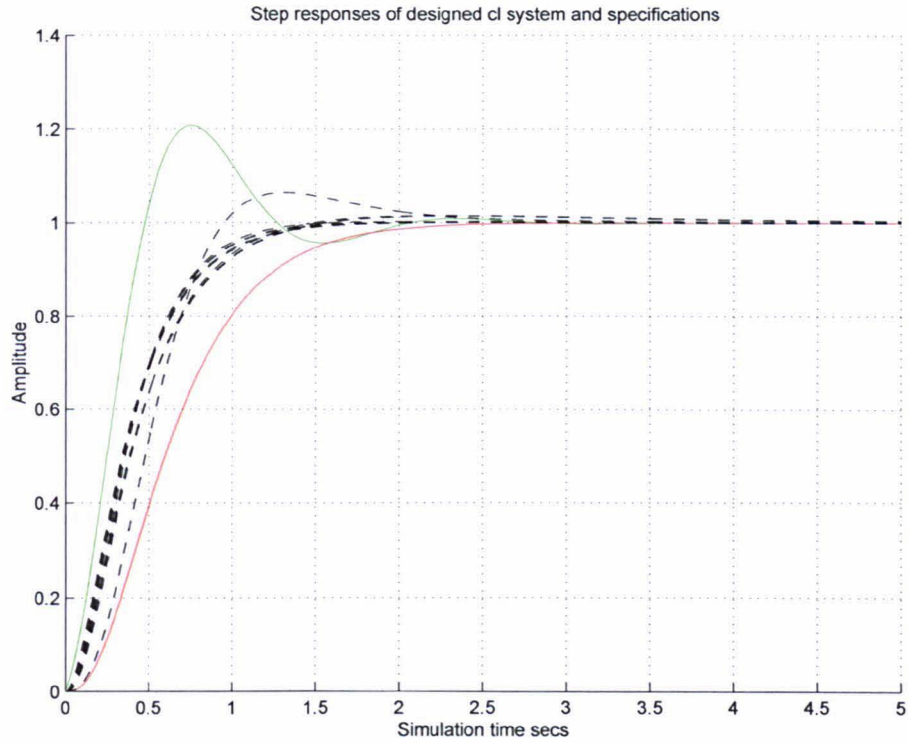


Figure 4.19: Step response of the system with the Pre-filter and controller in cascade

and

$$\log \beta(\omega) - \max_{p \in \mathcal{P}} \log \left| \frac{PG(j\omega)}{1 + L(j\omega)} \right| \geq \log |F(j\omega)| \quad (4.21)$$

*Illustration 4.7* This is the last design step of the example developed throughout this chapter. By using a straightforward model-matching procedure, the pre-filter was obtained as:

$$F(s) = \frac{24}{s^2 + 14s + 24}$$

Numerous simulations were carried for the system using the controller and the pre-filter designed above. The simulation results involving step and frequency responses of the designed system for various uncertain plant parameters are shown in figures 4.19 and 4.20 respectively. These show that the design specifications have been met.

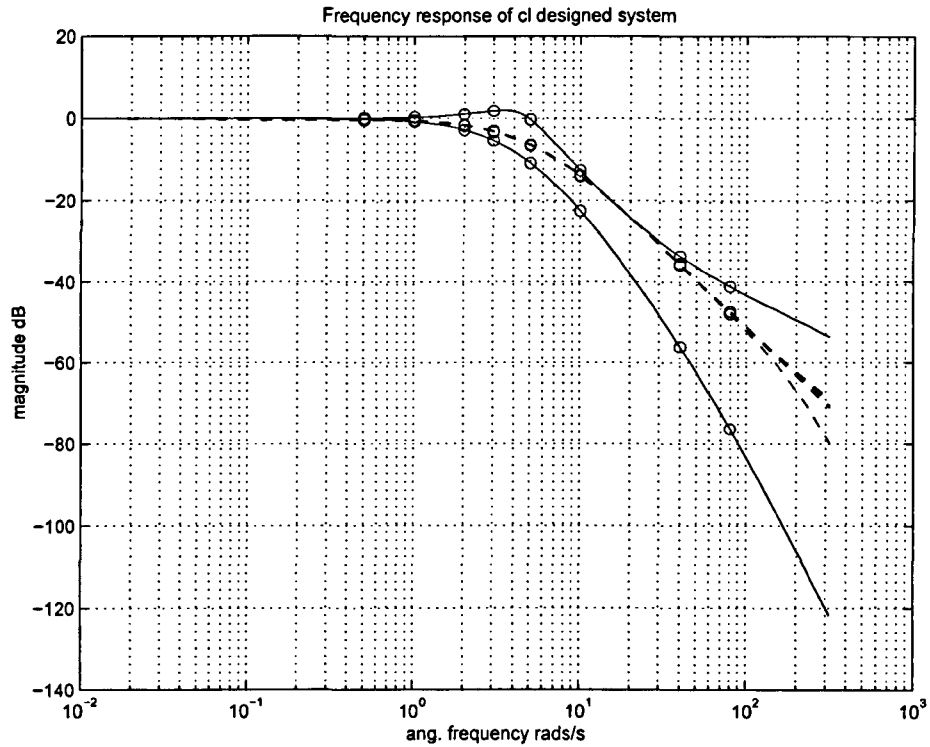


Figure 4.20: *Frequency response of the closed-loop system*

## 4.8 Multivariable system

Figure 4.21 shows a multivariable system with  $n \times n$  inputs and outputs. Plant  $G(s)$  and the controller  $K(s)$  are  $n \times n$  transfer function models.

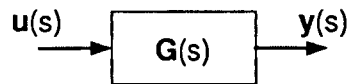


Figure 4.21: General Multivariable system

The problem with designing controllers for multivariable systems is the existence of *interaction* between the inputs and the outputs. The predominant method adapted by control system design engineers is to minimise or if possible completely eliminate any interactions between the various inputs and the outputs; this essentially results in a system where any one output will only respond to a specific input. Upon achieving such a setup it is possible to use the single input-single output design techniques for

designing the controller. Figure 4.22 shown a 2 x 2 multivariable system.

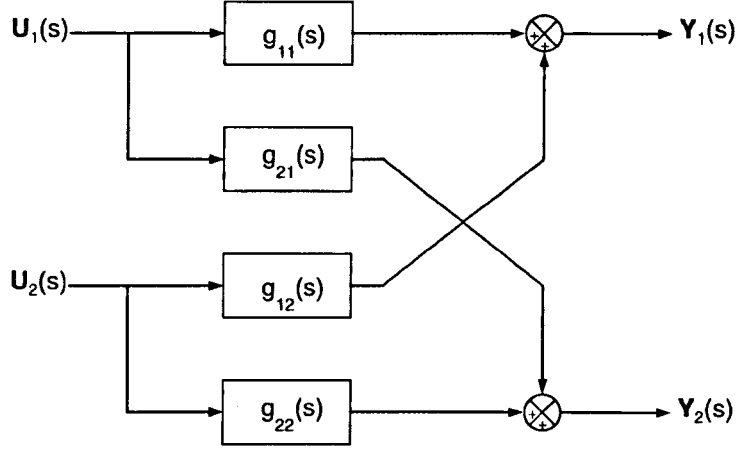


Figure 4.22: Expanded Multivariable system

This system can be shown as

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (4.22)$$

The interactions between inputs and outputs in equation 4.22 can be eliminated by forcing the elements  $g_{12}(s)$   $g_{21}(s)$  to 0, thus making  $\mathbf{G}(s)$  a diagonal matrix. Any system with a diagonal transfer function matrix will suffer no interactions thus effectively making it a set of individual **SISO** systems.

To extend this design method and design a controller using the QFT design technique [38] the design specification is defined by

$$a_{ij}(\omega) \leq |t_{ij}(j\omega)| \leq b_{ij}(\omega)$$

where  $t_{ij}(j\omega)$  is the  $(i, j)$  element of the closed-loop transfer function  $\mathbf{T}(s)$ .

Figure 4.23 can be written as

$$(I + GK)y = GKFr$$

assuming matrix  $G$  in the above equation is square and writing  $G^{-1}$  as  $\hat{G}$  we get

$$(\hat{G} + K)y = KFr$$

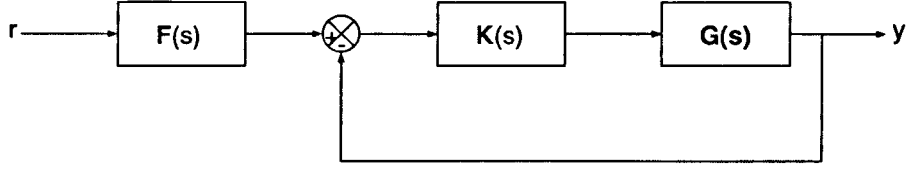


Figure 4.23: General feedback control system

Considering the function  $t_{\nu\nu}$  which represents the transfer function of the system from the  $\nu$ th input to  $\nu$ th output as shown in figure 4.24, if  $r_j = 0$  for  $j \neq \nu$ , then the  $\nu$  element of the vector  $KFr$  is given by

$$(KFr)_\nu = (KF)_{\nu\nu}r_\nu \quad (4.23)$$

$$= \sum_l k_{\nu l} f_{l\nu} r_\nu \quad (4.24)$$

The  $\nu$ th element of the vector  $(\hat{G} + K)y$  is given by

$$([\hat{G} + K]y)_\nu = \sum_l (\hat{g}_{\nu l} + k_{\nu l}) t_{l\nu} r_\nu \quad (4.25)$$

since  $y_l = t_{l\nu} r_\nu$ .

To achieve a diagonal system we impose the constraints  $k_{ij} = 0$  for  $i \neq j$ .

Equating equations 4.24 and 4.25 we get

$$r_\nu \sum_{l \neq \nu} \hat{g}_{\nu l} t_{l\nu} + r_\nu (\hat{g}_{\nu\nu} + k_{\nu\nu}) t_{\nu\nu} = k_{\nu\nu} f_{\nu\nu} r_\nu \quad (4.26)$$

Now by defining  $h_{ij} = \frac{1}{\hat{g}_{ij}}$ , eq 4.26 can be written as

$$t_{\nu\nu} = \frac{h_{\nu\nu} k_{\nu\nu} f_{\nu\nu}}{1 + h_{\nu\nu} k_{\nu\nu}} - \frac{h_{\nu\nu} d_{\nu\nu}}{1 + h_{\nu\nu} k_{\nu\nu}} \quad (4.27)$$

where

$$d_{\nu\nu} = \sum_{l \neq \nu} \frac{t_{l\nu}}{h_{\nu l}} \quad (4.28)$$

It can be observed from the above equation that  $t_{\nu\nu}$  now represents an output for a **SISO** system as shown in figure 4.24. Now that a multivariable system is represented by a set of **SISO** transfer function we can use the QFT method described in the previous chapters to design a controller.



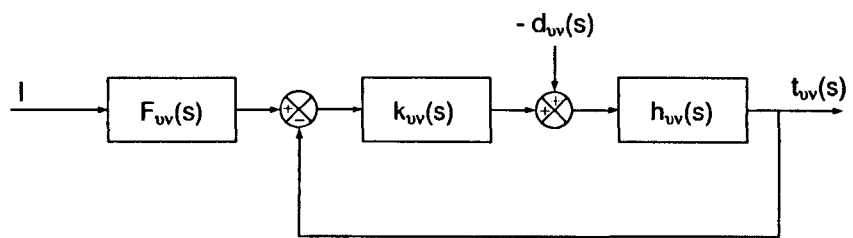


Figure 4.24: Expanded Multivariable system

# Chapter 5

## Loop-shaping

In chapter 4 the main steps of the QFT design methodology were described. This culminates in the loop-shaping design procedure, in which the nominal open-loop frequency response is shaped in the Nichols chart, so that the design meets the robust stability and performance specifications. This chapter describes two automatic optimisation-based loop-shaping methods.

The first method optimises the frequency response of the plant. In this formulation the optimisation variables consist of the open-loop magnitudes and phases at a discrete set of frequencies. The robust performance and robust stability bounds are formulated as linear inequalities, and form the constraints of a Linear Programming (LP) optimisation problem [6], which minimises the asymptotic open-loop gain of the system. To ensure that a realisable controller can be recovered from the solution to the optimisation problem, LP is augmented with additional analyticity and realisability constraints, obtained by discretising Bode's gain-phase integral relationship.

The second approach carries out the loop-shaping procedure by optimising over the parameters of certain types of fixed-structure controllers. Controllers which have been considered include PID, phase lead/lag and second-order pole/zero structures. The main idea of the approach is that fixing the phase of each of these type of controllers at two distinct frequencies, fixes its phase at *all* frequencies. With the dynamic part of the controller fixed, it is then straightforward to optimise its gain so that an appropriate performance index (asymptotic gain, nominal/worst-case cross-over frequency, nominal/worst-case closed-loop bandwidth, etc) are minimised, subject to the robust stability and robust performance QFT constraints defined graphically via the Horowitz

templates and  $U$ -contour bounds. Repeating this procedure over a two-dimensional discretised phase grid produces the optimal solution to the design problem.

## 5.1 Linear Programming

Linear programming is an optimisation method which minimises a *linear* cost function subject to a set of *linear* inequality or equality constraints. The method has a wide range of applications in many diverse areas, including Operations Research, Control Systems Design and Signal Processing. Linear problems of a very high dimension can be solved efficiently using a variety of different algorithms, including the Simplex algorithm and the more recent Interior point methods [15].

## 5.2 Bode integral

In this section we outline the first optimisation method. This is carried out directly in the frequency domain, i.e. the optimisation variables involve the magnitudes and phases of the open-loop system at a discrete-set of frequencies and no control structure is a-priori specified. To ensure that the optimised frequency response can be realised by an LTI dynamic system, we need to impose further constraints between the magnitude and phase variables. This is because the magnitude and phase response of an LTI system are intimately related via Bode's integral. Before formulating these additional constraints in a LP framework, we first examine the constraints arising from the robust stability and performance specifications ( $U$ -contour and Horowitz templates).

In order to formulate these constraints as linear inequalities, the robust stability/performance bounds need to be defined [19, 18]. Assuming the pre-filter is free from uncertainty, the open-loop needs to fulfill the following conditions to ensure robust stability at an arbitrary frequency  $\omega_i$ :

$$L(\omega_i) \in \Phi_i \Leftrightarrow \sup_{T \in \mathcal{T}} \Delta \log_{10} |T(j\omega_i)| \leq \log_{10} \frac{|T_u(j\omega_i)|}{|T_l(j\omega_i)|} \quad (5.1)$$

where

$$\Delta \log_{10} |T(j\omega_i)| := \max \log_{10} \left| \frac{T(j\omega_i)}{\tilde{T}(j\omega_i)} \right|, T(s), \tilde{T}(s) \in \mathcal{T} \quad (5.2)$$

in which  $T_u(j\omega_i)$  and  $T_l(j\omega_i)$  are the permissible upper and lower bounds magnitude for the closed-loop response and  $\mathcal{T}$  is the permissible closed-loop gain set at frequency  $\omega_i$ .

The design also needs to achieve robust disturbance rejection, i.e. to reject disturbances entering at the plant output for each (uncertain) plant in the specified uncertainty set. The condition below needs to be satisfied by the plant in order to achieve this objective:

$$20 \log_{10} \left| \frac{L(j\omega_i)}{1 + L(j\omega_i)} \right| \leq \mu$$

This equation defines a *restricted* region  $\Phi_{sens}$  in the Nichols chart at frequency  $\omega_i$ .

The optimum design for the plant under the robust stability conditions given by equations 5.1 and 5.2, is the one which satisfies the constraints using *as little feedback as possible*. This is in order to avoid overdesigning the system, e.g. by using larger bandwidth than really required. This can have adverse effects on the system stability due to unmodelled high frequency dynamics/parasitics and due to sensor noise amplification.

Under certain assumptions, Horowitz and Sidi [30] showed that,  $L_{opt}(j\omega)$  lies on the boundary of the region given by  $\Phi_i$  and  $\Phi_{sens}$ . However, the magnitude - phase characteristics of  $L_{opt}(j\omega)$  in the high frequency region is highly discontinues, which makes the design impractical. This issue can be solved by finding an approximation for  $L_{opt}(j\omega)$ . The approximate response  $L_{app}(j\omega)$  still needs to avoid the restricted regions by using sufficiently low gains. An alternative to the trial and error method used by Horowitz and Sidi to derive  $L_{app}(j\omega)$  was proposed by Bryant and Halikias [6] which uses the method of *Linear Programming*. The optimisation is carried out at a discrete set of frequencies  $\omega_i, i = 1, 2, \dots, N$ . To ensure that a realisable controller can be recovered from the optimal frequency response, the Bode phase-gain integral relation for  $L(s)$  is discretised at frequencies  $\omega_i$  and the resulting equalities are then imposed as additional constraints to the LP optimisation.

The Bode phase - gain integral is essentially a relation between the phase  $\arg L$  and the gain  $20 \log_{10} |L|$  of a system which is stable and minimum phase and has a pole/zero excess of at least two. This last condition is always satisfied by practical systems. The

derivation of the relation [18] is given in the next section.

### 5.2.1 Derivation

In order to establish Bode's gain-phase relation the following preliminary result is required:

**Theorem 5.1:-** Assume that  $L$  is proper,  $L$  and  $L^{-1}$  are analytic in  $\text{Re } s \geq 0$ , and  $L(0) > 0$ . Define  $G := \ln |L|$ . Then

$$\Re G = \ln |L|, \quad \Im G = \arg L$$

where  $\Re(\cdot)$  and  $\Im(\cdot)$  denote the real and imaginary parts respectively.  $G$  has the following properties:

- $G$  is analytic in some right half - plane containing the imaginary axis.
- $\Re G(j\omega)$  is an even function of  $\omega$  and  $\Im G(j\omega)$  is an odd function of  $\omega$ .
- $s^{-1}G(s)$  tends to zero uniformly on semicircles in the right half-plane as the radius tends to infinity, that is

$$\lim_{R \rightarrow \infty} \sup_{-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}} \left| \frac{G(Re^{j\theta})}{Re^{j\theta}} \right| = 0$$

**Proof:-** Since

$$G(Re^{j\theta}) = \ln |L(Re^{j\theta})| + j \arg L(Re^{j\theta})$$

and  $\arg L(Re^{j\theta})$  is bounded as  $R \rightarrow \infty$ , we have

$$\left| \frac{G(Re^{j\theta})}{Re^{j\theta}} \right| \rightarrow \frac{|\ln |L(Re^{j\theta})||}{R}.$$

Now  $L$  is proper, so for some  $c$  and  $k \geq 0$ ,

$$L(s) \approx \frac{c}{s^k} \quad \text{as} \quad |s| \rightarrow \infty$$

Thus

$$\begin{aligned} \left| \frac{G(Re^{j\theta})}{Re^{j\theta}} \right| &\rightarrow \frac{|\ln |L(Re^{j\theta})||}{R} \\ &= \frac{|\ln |\frac{c|}{R^k}||}{R} \\ &\rightarrow k \frac{\ln R}{R} \\ &\rightarrow 0. \end{aligned}$$

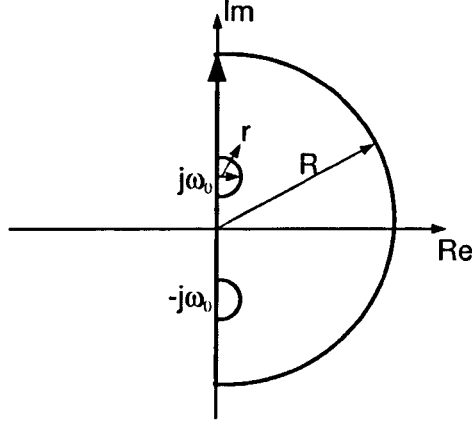


Figure 5.1: *Differentiability region*

**Lemma 5.1:** For each frequency  $\omega_0$

$$\Im G(j\omega_0) = \frac{2\omega_0}{\pi} \int_0^\infty \frac{\Re G(j\omega) - \Re G(j\omega_0)}{\omega^2 - \omega_0^2} d\omega.$$

**Proof:** Define the function:

$$\begin{aligned} F(s) &:= \frac{G(s) - \Re G(j\omega_0)}{s - j\omega_0} - \frac{G(s) - \Re G(j\omega_0)}{s + j\omega_0} \\ &= 2j\omega_0 \frac{G(s) - \Re G(j\omega_0)}{s^2 + j\omega_0^2}. \end{aligned}$$

Then  $F$  is analytic in the right half-plane and on the imaginary axis, except for poles at  $\pm j\omega_0$ . The integral of  $F$  around the Nyquist contour (which goes up the imaginary axis indenting to the right at the points  $-j\omega_0$  and  $j\omega_0$  along the semicircles of radius  $r$ , and then closing the contour by a large semicircle of radius  $R$  in the right half-plane) is zero (Cauchy's theorem). This integral equals the sum of six separate integrals corresponding to the three intervals on the imaginary axis, the smaller semicircles, and the large semicircles. Let  $I_1$  denote the sum of the three integrals on the imaginary axis.  $I_2$  the integral around the lower small semicircle,  $I_3$  around the upper small semicircle, and  $I_4$  around the large semicircle. It is shown that:

$$\begin{aligned}
\lim_{R \rightarrow \infty, r \rightarrow 0} I_1 &= 2\omega_0 \int_{-\infty}^{\infty} \frac{\Re G(j\omega) - \Re G(j\omega_0)}{\omega^2 - \omega_0^2} d\omega. \\
\lim_{r \rightarrow 0} I_2 &= -\pi \Im G(j\omega_0), \\
\lim_{r \rightarrow 0} I_3 &= -\pi \Im G(j\omega_0), \\
\lim_{R \rightarrow 0} I_4 &= 0.
\end{aligned}$$

The lemma follows immediately from these four equations and the fact that  $\Re G(j\omega)$  is even.

First consider,

$$I_1 = \int jF(j\omega) d\omega,$$

where the integral is over the set

$$[-R, -\omega_0 - r] \cup [-\omega_0 + r, \omega_0 - r] \cup [\omega_0 + r, R].$$

As  $R \rightarrow \infty$  and  $r \rightarrow 0$ , this set becomes the interval  $(-\infty, \infty)$ . Also,

$$jF(j\omega) = 2\omega_0 \frac{G(j\omega) - \Re G(j\omega_0)}{\omega^2 - \omega_0^2}.$$

Since

$$\frac{\Im G(j\omega)}{\omega^2 - \omega_0^2}$$

is an odd function, its integral over set equals zero, and we therefore get the required result.

Secondly,

$$\begin{aligned}
I_2 &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{G(-j\omega_0 + re^{j\theta}) - \Re G(j\omega_0)}{-j\omega_0 + re^{j\theta} - j\omega_0} jre^{j\theta} d\theta \\
&\quad - \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{G(-j\omega_0 + re^{j\theta}) - \Re G(j\omega_0)}{-j\omega_0 + re^{j\theta} + j\omega_0} jre^{j\theta} d\theta.
\end{aligned}$$

As  $r \rightarrow 0$ , the first integral tends to 0 while the second integral tends to:

$$[G(-j\omega_0) - \Re G(j\omega_0)] \quad j \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} d\theta = \pi \Im G(j\omega_0).$$

This proves the second integral relation for  $I_2$ . The proof for  $I_3$  is similar.

Finally,

$$I_4 = - \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} F(Re^{j\theta}) j Re^{j\theta} d\theta,$$

and so

$$|I_4| \leq \sup_{-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}} \left| \frac{2\omega_0 [G(Re^{j\theta}) - ReG(j\omega_0)]}{(Re^{j\theta})^2 + \omega_0^2} \right| R\pi.$$

Thus

$$|I_4| \rightarrow (\text{const}) \sup_{\theta} \frac{|G(Re^{j\theta})|}{R} \rightarrow 0.$$

This proves the result.

Rewriting the formula in the lemma in terms of  $L$  we get

$$\arg L(j\omega_0) = \frac{2\omega_0}{\pi} \int_0^\infty \frac{\ln |L(j\omega)| - \ln |L(j\omega_0)|}{\omega^2 - \omega_0^2} d\omega. \quad (5.3)$$

This is now manipulated to get the gain-phase formula.  $\square$

**Theorem 5.2:** For every frequency  $\omega_0$

$$\arg L(j\omega_0) = \frac{1}{\pi} \int_{-\infty}^\infty \frac{d \ln |L|}{dv} \ln \coth \frac{|v|}{2} dv. \quad (5.4)$$

where the integration variable  $v = \ln(\frac{\omega}{\omega_0})$ . This is called the *Bode phase-gain integral*.

**Proof:** Change variables of integration in 5.3 to get

$$\arg L(j\omega_0) = \frac{1}{\pi} \int_{-\infty}^\infty \frac{\ln |L(j\omega)| - \ln |L(j\omega_0)|}{\sinh v} dv.$$

Note that in the integral  $\ln |L|$  is really  $\ln |L(j\omega_0 e^v)|$  considered as a function of  $v$ . Now integrating by parts, from  $-\infty$  to 0 and from 0 to  $\infty$  gives:

$$\arg L(j\omega_0) = - \frac{1}{\pi} \left[ (\ln |L| - \ln |L(j\omega_0)|) \ln \coth \frac{v}{2} \right]_0^\infty \quad (5.5)$$

$$+ \frac{1}{\pi} \int_0^\infty \frac{d \ln |L|}{dv} \ln \coth \frac{v}{2} dv \quad (5.6)$$

$$+ \frac{1}{\pi} \left[ (\ln |L| - \ln |L(j\omega_0)|) \ln \coth \frac{-v}{2} \right]_0^\infty \quad (5.7)$$

$$+ \frac{1}{\pi} \int_{-\infty}^0 \frac{d \ln |L|}{dv} \ln \coth \frac{-v}{2} dv. \quad (5.8)$$

The first and third terms sum to zero.  $\square$



Equation 5.4 is called the Bode phase - gain integral. This equation is used in the rest of this chapter (in discretised form) to obtain linear inequalities which are then used for automatic loop-shaping.

### 5.2.2 Approximate Discretisation of Bode's integral

In this section we try to find a linear approximation relation between  $|L(j\omega)|$  and  $\phi(\omega)$  by discretising the Bode integral relation in equation 5.4. The Bode integral relation in equation 5.4 is rewritten here as:

$$\phi(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \frac{|\lambda|}{2} d\lambda \quad (5.9)$$

where

$$\lambda = \ln \frac{\zeta}{\omega}$$

and

$$\phi(\omega) = \arg L(j\omega)$$

The approximation of the Bode integral relation is done by discretising the equation. The infinite range of integration is divided into  $n + 1$  intervals, i.e.

$$(-\infty \lambda_n] \cup (\lambda_n \lambda_{n-1}] \cup \dots \cup (\lambda_2 \lambda_1] \cup [\lambda_1 \infty)$$

where  $\lambda_n < \lambda_{n-1} < \dots < \lambda_2 < \lambda_1$ . The  $\lambda_i$  are selected to cover adequately the bandwidth of the system. This approximation was suggested by [6, 38, 54].

Thus:

$$\phi(\omega) = \frac{1}{\pi} \int_{-\infty}^{\lambda_1} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (5.10)$$

$$+ \frac{1}{\pi} \sum_{j=1}^{n-1} \int_{\lambda_j}^{\lambda_{j+1}} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (5.11)$$

$$+ \frac{1}{\pi} \int_{\lambda_n}^{\infty} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (5.12)$$

$$(5.13)$$

The approximation is now carried out by assuming that the term  $d \ln |L(j\zeta)|/d\lambda$  is

constant in each interval. This gives:

$$\begin{aligned}\phi(\omega_k) &\cong \frac{1}{\pi} \frac{\ln\left(\frac{L_{j+1}}{L_j}\right)}{\ln\left(\frac{\omega_{j+1}}{\omega_j}\right)} \int_{-\infty}^{\ln \frac{\omega_1}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \\ &+ \frac{1}{\pi} \sum_{j=1}^{n-1} \frac{\ln\left(\frac{L_n}{L_{n-1}}\right)}{\ln\left(\frac{\omega_n}{\omega_{n-1}}\right)} \int_{\ln \frac{\omega_j}{\omega_k}}^{\ln \frac{\omega_{j+1}}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \\ &+ \frac{1}{\pi} \frac{\ln\left(\frac{L_2}{L_1}\right)}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \int_{\ln \frac{\omega_n}{\omega_k}}^{\infty} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda\end{aligned}$$

for  $k = 0, 2, \dots, n-1$ . The integrals in the above expression can be calculated to any desired degree of accuracy via numerical integration. On noting that  $\ln(L_{i+1}/L_i) = \ln(L_{i+1}) - \ln(L_i)$ , we can see that each  $\phi_k = \phi(\omega_k)$  is a linear combination of the  $\ln(L_i)$ 's, i.e.

$$\phi_k = \sum_{i=1}^n a_{ki} M_i \quad k = 2, 3, \dots, n-1$$

for some constants  $a_{ki}$ . Here,

$$M_i = 20 \log_{10} |L(j\omega_i)|$$

represent the open-loop gain at frequencies  $\omega_i$  in dB's. A full derivation is included in Appendix A.

Illustration 5.1 The phase reconstruction method explained above is illustrated using two transfer functions in equations (5.14) and (5.15).

$$L_1(s) = \frac{s+5}{s^2+10s+1} \quad (5.14)$$

$$L_2(s) = \frac{s+3}{s^4+6s^3+7s^2+6s+1} \quad (5.15)$$

The MATLAB file used to generate the approximate and true phases is included in appendix C. The results are shown in figures 5.2 and 5.3. These show that this procedure reconstructs the phase highly accurately when a reasonable number of gains are used.

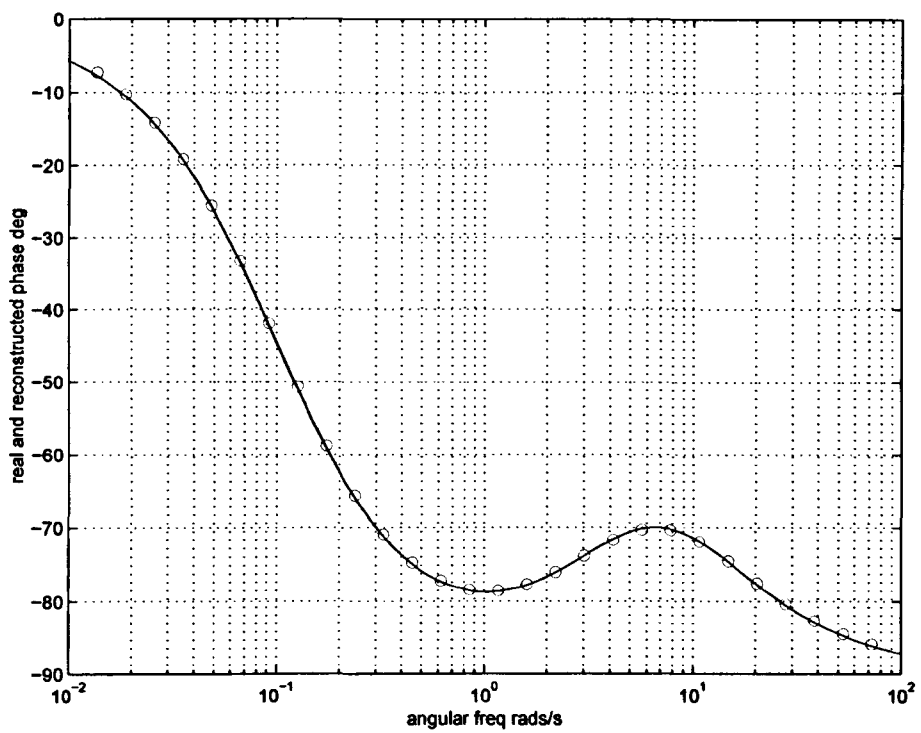


Figure 5.2: *Phase reconstruction for first transfer function*

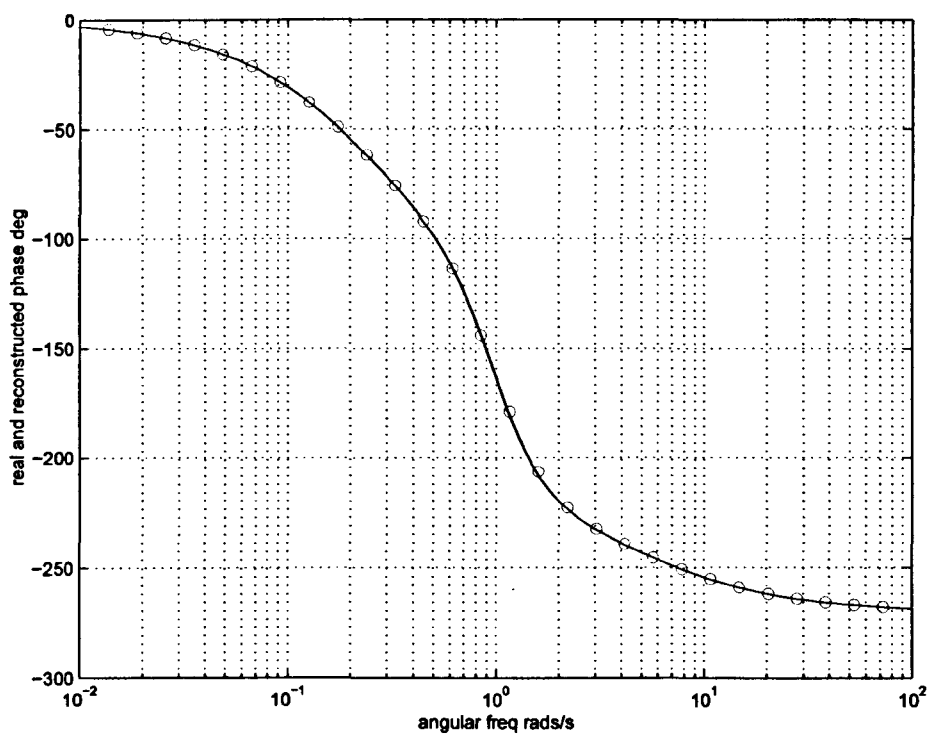


Figure 5.3: *Phase reconstruction for second transfer function*

### 5.3 Analytic constraints

Imposing just Bode constraints to the linear programme will result to a controller which although realisable, may have a highly discontinuous frequency response. In order to obtain a smooth response, we need to impose additional constraints to the LP which ensure that  $\ln |L(s)|$  is differentiable on the imaginary axis. Such constraints were proposed by Unstead and MacLeod in the form of cubic polynomials [54].

A function  $f(s)$  of a complex variable  $s$  is said to be *analytic* in a domain  $D$  of the complex plane if  $f(s)$  is defined and the derivative exists at all points in  $D$ . Given that the limit exists, the derivative of  $f$  at a point  $s_0$  is written as  $f'(s)$  and is defined as:

$$f'(s_0) = \lim_{s \rightarrow s_0} \frac{f(s) - f(s_0)}{s - s_0} \quad (5.16)$$

The function is said to be analytic, or differentiable at a point  $s = s_0$  in  $D$  if  $f(s)$  is analytic in a neighborhood of  $s_0$ . Based on the above definition the differentiability constraints are derived, on the basis that  $\ln L(s)$  must be analytic on the  $j\omega$  axis. The constraints ensure that the above limit is the same when  $s_0$  is approached from both directions along the imaginary axis.

Let  $\hat{L}(s) = \ln L(s)$ . Then

$$\hat{L}(j\omega) = \ln |L(j\omega)| + j \arg L(j\omega) = u(\omega) + jv(\omega)$$

and using above equation we get

$$\begin{aligned} \hat{L}'(j\omega) &= \lim_{\epsilon \rightarrow 0} \frac{\hat{L}[j(\omega + \epsilon)] - \hat{L}[j(\omega)]}{j(\omega + \epsilon) - j\omega} \\ &= \lim_{\epsilon \rightarrow 0} \frac{u(\omega + \epsilon) + jv(\omega + \epsilon) - u(\omega) - jv(\omega)}{j\epsilon} \\ &= \lim_{\epsilon \rightarrow 0} \left[ \frac{v(\omega + \epsilon) - v(\omega)}{\epsilon} \right] - j \left[ \frac{u(\omega + \epsilon) - u(\omega)}{\epsilon} \right] \end{aligned}$$

In this relation  $u(\omega)$  and  $v(\omega)$  are approximated for an interval  $[\omega_i, \omega_{i+1}]$  as cubic polynomials in terms of  $u_i(\omega)$  and  $v_i(\omega)$  where:

$$u_i(\omega) = p_{i1}\omega^3 + p_{i2}\omega^2 + p_{i3}\omega + p_{i4} \quad (5.17)$$

$$v_i(\omega) = q_{i1}\omega^3 + q_{i2}\omega^2 + q_{i3}\omega + q_{i4} \quad (5.18)$$

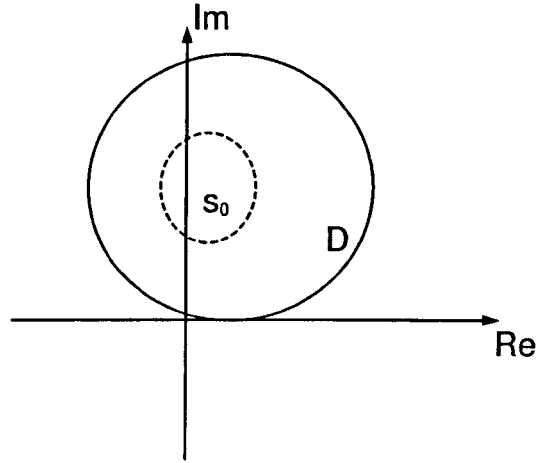


Figure 5.4: *Differentiability region*

In order to formulate the constraints, equation 5.17 and 5.18 needs to be represented as a matrix relation between the magnitude and the phase of the system. The derivation of these linear relations is given in appendix B.

A Linear Programme may now be formulated in terms of the unknown variables  $L_i$  and  $\phi_i$  which represent the nominal open-loop gain and phase at a discrete set of frequencies. The (linear) constraints of the LP are: (i) Linear approximations of the  $U$ -contour and the Horowitz templates (at specified phase intervals) which define the robust stability and robust performance specifications; (ii) Equality constraints arising from the discretised Bode gain/phase integral relation, which ensure that a realisable LTI controller can be recovered, and (iii) Equality analyticity constraints which ensure that the solution will exhibit a smooth frequency response. The objective function of the LP which is minimised can be taken to be the asymptotic open loop gain or some other measure of the open-loop gain response which limits system over-design.

## 5.4 Fixed-structure controllers

### 5.4.1 Introduction

In this section a simple optimisation algorithm is proposed for designing fixed-structure controllers for highly-uncertain systems. This allows the design of robust controllers widely used in industry, such as PID (Proportional, Integral, Derivative) and phase-

lead/phase-lag filter-structure. The method can be used to automate the loop-shaping step of the QFT design procedure and guarantees robust stability and performance to the feedback loop for all parameters in the plant's uncertainty set. To avoid over-designing the system, the algorithm can be used to minimize either the asymptotic gain, the open-loop cross-over frequency or the 3-dB bandwidth of the closed-loop system (nominal or worst-case). The resulting algorithm for each structure is easy to implement and relies only on simple magnitude and phase calculations.

### 5.4.2 Problem description

The QFT design procedure is based on the two-degree of freedom feedback configuration shown in Figure 1. In this diagram  $G(p, s)$  denotes the uncertain plant, while  $K(s)$  and  $F(s)$  denote the feedback compensator and pre-filter, respectively, which are to be designed. Note that model uncertainty is described by the  $r$ -parameter vector  $p \in \mathcal{P} \subseteq \mathcal{R}^r$  taking values in the set  $\mathcal{P}$ ; it is further assumed that  $G(p, s)$  has the same number of right-half-plane (RHP) poles for all  $p \in \mathcal{P}$ . Translating the uncertainty into the frequency domain, gives rise to the plant's "uncertainty templates" which are the sets:

$$\mathcal{G}_\omega = \{G(p, j\omega) : p \in \mathcal{P}\}$$

For each fixed frequency  $\omega$ ,  $\mathcal{G}_\omega$  defines a "fuzzy region" on the Nichols chart which describes the uncertainty of the plant at frequency  $\omega$  in terms of magnitude (in dB's) and phase (in degrees). For design purposes, we construct  $N$  uncertainty templates corresponding to a discrete set of frequencies  $\Omega = \{\omega_1, \omega_2, \dots, \omega_N\}$  chosen to cover adequately the system's bandwidth.

The robust performance objectives of the design include good tracking of reference input  $r(s)$  and good attenuation of the disturbance signal  $d(s)$  entering at the system's output, despite the presence of uncertainty. The robust tracking objectives are captured by the set of inequalities:

$$\max_{p \in \mathcal{P}} \Delta \left| \frac{G(p, j\omega_i)K(j\omega_i)}{1 + G(p, j\omega_i)K(j\omega_i)} \right|_{\text{dB}} \leq \delta(\omega_i) := B_u(\omega_i)|_{\text{dB}} - B_l(\omega_i)|_{\text{dB}}$$

for each  $i = 1, 2, \dots, N$ , i.e. if, for each frequency  $\omega_i$ , the maximum variation in closed loop gain as  $p \in \mathcal{P}$  does not exceed the maximum allowable spread in specifications  $\delta(\omega_i)$ , typically specified via two appropriate magnitude frequency responses  $B_u(\omega) =$

$|B_u(j\omega)|$  and  $B_l(\omega) = |B_l(j\omega)|$ . Note that it is not necessary to bound the actual gain (but only the gain spread) since we assume that, (i) no uncertainty is associated with the feedback controller  $K(s)$ , and (ii) the pre-filter  $F(s)$  can provide arbitrary scaling to the closed-loop gain at every frequency.

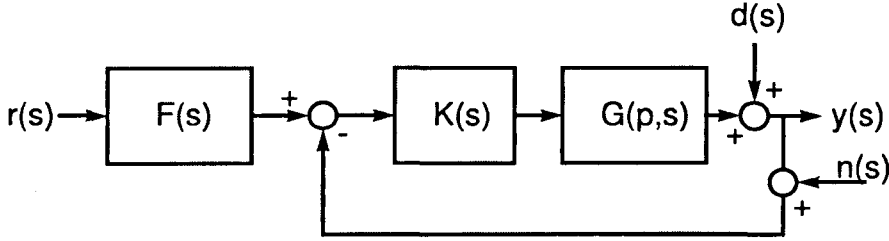


Figure 5.5: Feedback Configuration

The robust disturbance-rejection objective is satisfied by bounding the sensitivity function, i.e. by imposing constraints of the form

$$\max_{p \in \mathcal{P}} \left| \frac{1}{1 + G(p, j\omega_i)K(j\omega_i)} \right| \leq D(\omega_i)$$

for a (subset) of the design frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$  (normally in the low-frequency range). Again these are typically specified via an appropriate magnitude frequency-response  $D(\omega) = |D(j\omega)|$ .

Robust stability is enforced by ensuring that: (i) no unstable pole-zero cancellations occur between the plant and the controller (for every  $p \in \mathcal{P}$ ), (ii) the nominal open-loop frequency response  $L_o(j\omega) = G(p_o, j\omega)K(j\omega)$  (defined for any  $p_o \in \mathcal{P}$ ) does not cross the  $-1$  point, i.e. the  $(-180^\circ, 0)$  point on Nichols chart), and makes a total number of (anti-clockwise) encirclements around it equal to the number of unstable poles of  $L_o(s) = G(p_o, s)K(s)$ , and (iii) That no (perturbed) open-loop response crosses the  $-1$  point, i.e.

$$-1 \notin \bigcup_{\omega \in \mathcal{R}} K(j\omega)G_\omega$$

Note that condition (i) is automatically satisfied if  $K(s)$  is restricted to be stable and minimum-phase, while conditions (ii) and (iii) can be easily tested graphically [13], [12]. In practice, a more severe condition than (iii) is imposed: To establish a minimum amount of damping, it is required that the nominal open-loop frequency response

does not penetrate a closed contour in the Nichols chart (universal high-frequency  $\mathcal{U}$ -contour); this is constructed from an appropriate  $M$ -circle and information about high-frequency gain uncertainty of the plant [30]. Formulation of robust stability via the  $\mathcal{U}$ -contour assumes that at high-frequencies the phase-uncertainty spread of the system is minimal, an assumption which is reasonable for most systems subject to *parametric* model uncertainty. If this assumption fails (or if model uncertainty is in part unstructured) the  $\mathcal{U}$ -contour must be replaced by a set of frequency-dependent closed-templates containing the critical point. This does not affect significantly the proposed method, although for simplicity it is assumed that robust-stability specifications can be formulated via the  $\mathcal{U}$ -contour.

The robust tracking and disturbance rejection objectives have been formulated as gain inequalities of the closed-loop transfer functions (sensitivity and complementary sensitivity) at the design frequencies. For the purposes of QFT design, these inequalities must be translated into constraints on the nominal open-loop response  $L_o(j\omega)$ . This procedure results into a number of contours (“Horowitz tracking templates”  $f_i^t(\phi)$  and “Horowitz disturbance-rejection templates”  $f_i^d(\phi)$ ) for each frequency  $\omega_i$ ,  $i = 1, 2, \dots, N$ ; these are functions of the phase variable  $\phi \in (-360^\circ, 0^\circ]$ . Thus, robust tracking is satisfied at frequency  $\omega_i$  if and only if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$  where  $\arg L_o(j\omega_i) = \phi_i$ ; similarly, robust disturbance-rejection is attained at frequency  $\omega_i$  if and only if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi)$ . The robust-performance templates (Horowitz-tracking and disturbance-rejection) can be easily constructed (within an arbitrary gain tolerance and for a discretised phase-grid) using a simple bisection algorithm. This method uses the uncertainty templates of the plant defined at the design frequencies, normally obtained by “gridding” the uncertainty parameters (which may be computationally expensive), although more sophisticated methods have been proposed, e.g. [13], [12].

In conclusion, assuming that the condition prohibiting unstable pole/zero cancellations between the plant and the controller is independently verified, the following conditions guarantee robust-stability and performance:

- The winding number of the nominal open-loop system  $L_o(j\omega)$  around the  $-1$  point is equal to the number of RHP poles of  $L_o(s)$ .



- The nominal open-loop frequency response  $L_o(j\omega)$  does not penetrate the  $\mathcal{U}$  contour.
- The following inequalities are satisfied for all  $i = 1, 2, \dots, n$ :

$$|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$$

and

$$|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi_i)$$

in which  $\phi_i = \arg L_o(j\omega_i)$ ; these inequalities correspond to the robust tracking and robust disturbance-rejection specifications respectively.

The thesis presents a novel algorithm for designing fixed-structure controllers which satisfy the QFT constraints and minimize a measure of system “over-design” (asymptotic gain, cross-over frequency, closed-loop bandwidth). In section 5.4.3 the QFT constraints are formulated in the form of a feasibility programme. Section 5.4.4 outlines an optimization algorithm which can be used to design simple fixed-structure controllers (PID, phase-lead/lag, second-order) in the QFT framework. The algorithm is illustrated in section 5.4.5 with simple design examples, while the main conclusions appear in section 5.4.6.

### 5.4.3 Formulation of QFT constraints as frequency-dependent inequalities

In this section the QFT robust stability and performance constraints are first formulated as a feasibility programme. This leads to an optimisation algorithm for carrying out optimal QFT designs using a family of simple fixed-structure compensators. This is in contrast to other approaches (e.g. the method presented in the previous section) which optimise the open-loop response of the system in the frequency-domain and subsequently fit a (potentially high-order) compensator.

As was shown in the last section, the QFT robust-stability and performance objectives can be translated to graphical constraints on the Nichols chart. The constraints associated with robust-performance (“Horowitz-tracking” and “Horowitz disturbance-

rejection” templates) correspond to open contours, i.e. they split the Nichols chart in two regions (for each design frequency), the high and low-gain region. To meet the tracking or disturbance-rejection objective, each nominal open-loop frequency-response point  $L_o(j\omega_i)$  must be placed on the high-gain region of the contour, i.e. forced to satisfy the inequality  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$  (tracking) or  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi_i)$  (disturbance rejection), where  $\arg L_o(j\omega_i) = \phi_i$ . In contrast, the robust-stability template ( $\mathcal{U}$ -contour) is a closed contour containing the critical point. The construction of the  $\mathcal{U}$ -contour and the translation of the  $M$ - circle to the Nichols plot is described in section 4.2.1.

The ultimate objective of this section is to characterise the regions of the Nichols chart in which the open-loop frequency response point  $L_o(j\omega_i)$  can lie in order to satisfy the robust stability and performance constraints. To this purpose define the composite function

$$f_i^m(\phi) = \max\{f_i^t(\phi), f_i^d(\phi)\}$$

where the maximum is taken point-wise in  $\phi \in (-360^\circ, 0^\circ]$ . Further define,

$$\begin{aligned} f_i(\phi) &= f_i^m(\phi) && \text{for } \phi \leq \phi_l, \\ &= \max\{f_i^m(\phi), \mathcal{U}^+(\phi)\} && \text{for } \phi_l < \phi < \phi_h \\ &= f_i^m(\phi) && \text{for } \phi \geq \phi_h \end{aligned}$$

Also let  $\Phi_i = \{\phi : \phi_l < \phi < \phi_h, f_i^m(\phi) \leq \mathcal{U}^-(\phi)\}$ . Then, the robust stability and performance constraints at frequency  $\omega_i$  are satisfied if and only if  $L_o(j\omega_i) \in R_i \cup S_i$ , where

$$R_i = \{L_o(j\omega_i) : |L_o(j\omega_i)|_{\text{dB}} \geq f_i(\phi), \phi = \arg L_o(j\omega_i)\}$$

and

$$S_i = \{L_o(j\omega_i) : f_i(\phi) \leq |L_o(j\omega_i)|_{\text{dB}} \leq \mathcal{U}^-(\phi), \phi = \arg L_o(j\omega_i) \in \Phi_i\}$$

An illustration of the region  $R_i \cup S_i$  is given in Figure 3. Note that in practice, when a performance constraint is active, we typically have  $\Phi_i = S_i = \emptyset$ . This is because performance objectives are normally specified at low frequencies, rarely exceeding the closed-loop bandwidth of the system. However, the present formulation makes it possible to take into account “unconstrained” design frequencies, i.e. frequencies at which no performance inequalities apply. For such a frequency  $\omega_i$  we would have

$f_i^m(\phi) = -\infty$  and hence  $\Phi_i = (\phi_l, \phi_h)$ ,  $S_i = \{L_o(j\omega_i) : |L_o(j\omega_i)|_{\text{dB}} \leq \mathcal{U}^-(\phi), \phi = \arg L_o(j\omega_i) \in (\phi_l, \phi_h)\}$  (i.e. the region below the  $\mathcal{U}$  contour), while  $R_i \cup S_i$  would represent the region outside the  $\mathcal{U}$ -contour.

The conditions that guarantee the robust-stability and robust-performance specifications at the design frequencies can now be summarised by the following two graphical tests:

- The winding number of the nominal open-loop system  $L_o(j\omega)$  around the  $-1$  point is equal to the number of RHP poles of  $L_o(s)$ .
- For each frequency  $\omega_i$ ,  $L_o(j\omega_i) \in R_i \cup S_i$ .

Again, it is assumed that no unstable pole/zero cancellations occur between the controller and  $G(p, s)$  for every  $p \in \mathcal{P}$ , a condition which must be checked independently. Of course, similarly to any QFT-based method, these tests do not really guarantee that  $L_o(j\omega)$  does not enter the  $\mathcal{U}$  contour at frequencies other than the design frequencies. This, however, does not cause a problem in practice, provided a reasonably large set of design frequencies is selected near cross-over, or, alternatively, by slightly tightening the specifications by means of an appropriate tolerance.

#### 5.4.4 Optimisation algorithm

In this section we outline an optimisation algorithm for designing fixed-structure compensators of certain types subject to the QFT constraints developed earlier. Every design (i.e. loop shaping of  $L_o(j\omega)$ ) which satisfies the two graphical tests of the last section is in principle “admissible”, i.e. satisfies the robust-stability and the robust-performance objectives. Since in general many different designs may be admissible, we require a method of classifying them by formulating an appropriate optimisation criterion. Adopting the arguments of Horowitz and Sidi [30], [29], such a criterion must penalise the “over-design” of the system, e.g. an unnecessarily high closed-loop bandwidth, since this increases the “cost of feedback” in terms of sensor-noise amplification and potential instability due to high-frequency un-modelled dynamics/parasitics. Appropriate “cost functions” to be minimised include the following quantities:

- Open-loop cross-over frequency (nominal or worst-case).

- Closed-loop bandwidth (nominal or worst-case).
- Asymptotic open-loop gain.
- A measure of the excess gain-bandwidth area which can be expressed as the integral

$$A(\omega_1, \omega_2) = \int_{\omega_1}^{\omega_2} \log |K(j\omega)| d\omega$$

where  $[\omega_1, \omega_2]$  is an appropriate frequency interval.

Each of the above measures can be calculated in a straightforward manner from the frequency response of the system. For example, the open-loop cross-over frequency corresponds to the point where the open-loop frequency response crosses the 0 dB line on the Nichols chart. The closed-loop bandwidth is the frequency where the closed loop gain of the system is  $1/\sqrt{2}$  (-3 dB approximately). To calculate the closed-loop bandwidth graphically let  $L = r \exp j\phi$  be the open loop response and set:

$$\frac{|L|}{|1 + L|} = \frac{1}{\sqrt{2}} \Rightarrow \frac{r^2}{1 + r^2 + 2r \cos \phi} = \frac{1}{2}$$

This leads to the quadratic equation  $r^2 - 2r \cos \phi - 1 = 0$  whose only admissible solution is  $r = \cos \phi + \sqrt{\cos^2 \phi + 1}$ . Thus the closed-loop bandwidth of the system is the frequency at which the open-loop frequency response crosses the curve:

$$N(\phi) = 20 \log_{10} \left( \cos \phi + \sqrt{\cos^2 \phi + 1} \right)$$

on the Nichols chart, where  $\phi$  denotes open-loop phase. The curve  $N(\phi)$  is plotted in Figure 5.6 over the phase interval  $(-360^\circ, 0^\circ]$ . Finally, the excess gain-bandwidth measure  $A(\omega_1, \omega_2)$  may be easily calculated by numerical integration in terms of the controller parameters.

Note that the open-loop response of most systems encountered in practice crosses the 0 dB line (or curve  $N(\phi)$ ) only once. An important exception consists of systems with lightly-damped modes (e.g. flexible structures) exhibiting multiple “resonance” peaks. In such cases we simply define the cross-over frequency (or closed-loop bandwidth) as the lowest frequency at which crossing occurs. We also define the “worst-case” cross-

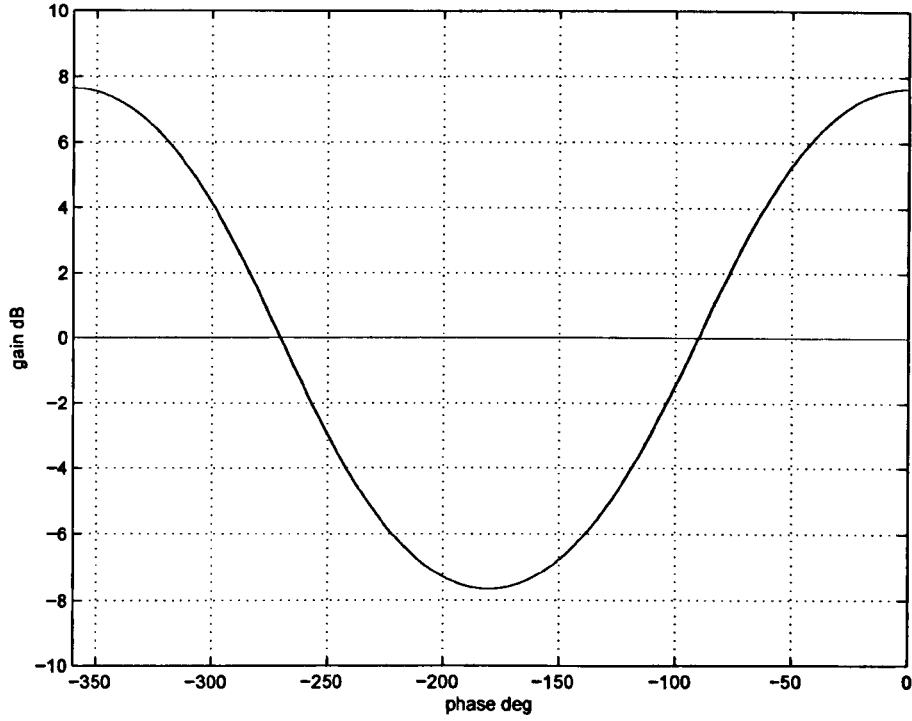


Figure 5.6: Curve  $N(\phi)$  used to calculate closed-loop bandwidth

over frequency or closed-loop bandwidth as the largest frequency among all uncertain frequency responses (contained in the uncertainty template set) crossing the 0 dB line or curve  $N(\phi)$ , respectively. Note that all these optimization measures can be easily calculated from the frequency response of the system, possibly using interpolation techniques if high accuracy is required.

The algorithm presented here may be used to automate the loop-shaping step of the QFT design algorithm. This is the most demanding step of the QFT design procedure [6], for which significant research effort has been devoted in the recent literature, e.g. the approach of [57] based on Youla's parametrization and linear programming, the approach of [15], [61] which extends the results of [16] to the robust QFT framework, techniques which rely on Bode's gain-phase integral to impose controller realizability constraints [22], [51], [6], etc.

The types of compensators considered here are listed below. Note that some of these must be used under appropriate relative-degree assumptions satisfied by the transfer function of the plant.

1. PID:  $K(s) = k_p + k_d s + \frac{k_i}{s}$  and PDD<sup>2</sup>:  $K(s) = k_1 + k_2 s + k_3 s^2$ .

2. First-order lead/lag:  $K(s) = \frac{k(s+b)}{s+a}$

3. Second-order with complex-poles (zeros):  $K(s) = \frac{1}{s^2+2\zeta\omega_n s+\omega_n^2}$  (or  $s^2+2\zeta\omega_n s+\omega_n^2$ ).

PID and phase lead/lag controllers are widely used in industry because they are simple and easy to tune. Thus optimal controllers of the first two types may provide simple solutions to robust control designs based on the QFT method. Note also that every rational controller of arbitrary complexity can be constructed from cascade interconnections of controllers in (2) and (3) above. Thus, it is possible to improve the design continuously by building higher-order controllers in a step-by-step procedure: At each step the optimisation algorithm is carried out (for a specified controller structure) and the resulting optimal controller  $K(s)$  is accumulated into the nominal open-loop system by redefining  $L_o(s) \leftarrow L_o(s)K(s)$ . This process may continue until a satisfactory design is obtained, or until the cost fails to decrease significantly. Of course, the controller resulting from this procedure will not, in general, be optimal over the higher-order controller set!

The proposed algorithm is based on the fact that fixing the phase of the compensator at two distinct frequencies determines the compensator uniquely *up to scaling*. Thus, the phase-response of the nominal open-loop system is also completely determined, and only a simple calculation is needed to determine the minimum amount of gain required to meet the QFT robust stability and performance specifications (if these are feasible). Geometrically, this corresponds to shifting the frequency response of  $L_o(s)$  vertically in the Nichols chart by the minimum gain required to place the the points  $L_o(j\omega_i)$  in the  $R_i \cup S_i$  regions while simultaneously satisfying the Nyquist stability encirclement criterion. Repeating this procedure for all possible phase combinations (suitably discretised) will eventually produce the optimal design (if one exists). Next, we consider each controller type in turn:

### ***PID and PDD<sup>2</sup> controllers:***

We first consider the classical PID controller, specified by three parameters  $k_i$ ,  $k_d$  and  $k_p$  corresponding to the integral, derivative and proportional gain, respectively.

**Theorem 5.3:** (i) Let  $K(s) = k_p + k_d s + \frac{k_i}{s}$  with  $k_p$ ,  $k_d$ ,  $k_i$  real parameters. Suppose

that  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  where  $\omega_i \neq \omega_j$ . Then the matrix:

$$A_{ij} = \begin{pmatrix} 1 & -\frac{1}{\omega_i^2} & -\frac{\tan(\psi_i)}{\omega_i} \\ 1 & -\frac{1}{\omega_j^2} & -\frac{\tan(\psi_j)}{\omega_j} \end{pmatrix}$$

has full (row) rank. Let  $(V^{ij}) = [V_1^{ij} \ V_2^{ij} \ V_3^{ij}]' \in \mathcal{R}^3$  be a (real) non-zero vector in the (one-dimensional) kernel of  $A_{ij}$ . Then,

$$\begin{pmatrix} k_d \\ k_i \\ k_p \end{pmatrix} = \lambda \begin{pmatrix} V_1^{ij} \\ V_2^{ij} \\ V_3^{ij} \end{pmatrix} := \lambda \begin{pmatrix} \frac{\omega_i \tan \psi_i - \omega_j \tan \psi_j}{\omega_i^2 - \omega_j^2} \\ \frac{\omega_i \omega_j (\omega_j \tan \psi_i - \omega_i \tan \psi_j)}{\omega_i^2 - \omega_j^2} \\ 1 \end{pmatrix}$$

where  $\lambda$  is an arbitrary real constant. Moreover, the gain and phase of the controller at any frequency  $\omega$  is given by:

$$\begin{aligned} |K(j\omega)| &= |\lambda| \sqrt{1 + \left( \frac{\omega(\omega_i \tan \psi_i - \omega_j \tan \psi_j)}{\omega_i^2 - \omega_j^2} - \frac{\omega_i \omega_j (\omega_j \tan \psi_i - \omega_i \tan \psi_j)}{\omega(\omega_i^2 - \omega_j^2)} \right)^2} \\ &:= |\lambda| C_{ij}(\omega) \end{aligned}$$

and

$$\arg K(j\omega) := \psi(\omega) = \arctan \left( \frac{\omega(\omega_i \tan \psi_i - \omega_j \tan \psi_j)}{\omega_i^2 - \omega_j^2} - \frac{\omega_i \omega_j (\omega_j \tan \psi_i - \omega_i \tan \psi_j)}{\omega(\omega_i^2 - \omega_j^2)} \right)$$

respectively.

(ii) If the controller gains  $k_p$ ,  $k_i$  and  $k_d$  are restricted to be non-negative, then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  are feasible if the two scalars  $V_1^{ij}$ ,  $V_2^{ij}$  are positive. In this case,

$$\phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{\omega_k(\omega_i \tan \psi_i - \omega_j \tan \psi_j)}{\omega_i^2 - \omega_j^2} - \frac{\omega_i \omega_j (\omega_j \tan \psi_i - \omega_i \tan \psi_j)}{\omega_k(\omega_i^2 - \omega_j^2)} \right)$$

where we have defined

$$\phi_k = \arg L_o(j\omega_k)$$

Also,  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $|\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \notin [\phi_l \ \phi_h]$ .
- $|\lambda| \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \ \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)} \geq |\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  or  $|\lambda| \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \ \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C_{ij}(\omega_k)$  is defined in part (i).

**Proof:** (i) The frequency response of the PID controller is given as

$$K(j\omega) = k_p + jk_d\omega - j\frac{k_i}{\omega}$$

with gain and phase

$$|K(j\omega)| = \sqrt{k_p^2 + \left(k_d\omega - \frac{k_i}{\omega}\right)^2} \quad \text{and} \quad \arg K(j\omega) = \arctan\left(\frac{k_d\omega - \frac{k_i}{\omega}}{k_p}\right)$$

respectively. Now suppose we fix  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two frequencies  $\omega_i \neq \omega_j$ . Then,

$$\begin{aligned} k_d - \frac{k_i}{\omega_i^2} - \frac{k_p \tan(\psi_i)}{\omega_i} &= 0 \\ k_d - \frac{k_i}{\omega_j^2} - \frac{k_p \tan(\psi_j)}{\omega_j} &= 0 \end{aligned}$$

which can be written in matrix form as:

$$\begin{pmatrix} 1 & -\frac{1}{\omega_i^2} & -\frac{\tan(\psi_i)}{\omega_i} \\ 1 & -\frac{1}{\omega_j^2} & -\frac{\tan(\psi_j)}{\omega_j} \end{pmatrix} \begin{pmatrix} k_d \\ k_i \\ k_p \end{pmatrix} = 0$$

Clearly,  $\text{Rank}(A_{ij}) = 2$ , since  $\omega_i \neq \omega_j$  and thus the controller parameter vector is constrained to lie in the one-dimensional subspace  $\text{Ker}(A_{ij})$ . Writing  $\text{Ker}(A_{ij}) = \lambda[V_1^{ij} \ V_2^{ij} \ V_3^{ij}]'$  gives the required expressions for  $k_d$ ,  $k_i$  and  $k_p$  from which the magnitude and phase expressions of  $K(j\omega)$  follow after some simple algebra. (ii) It is clear that when the controller gains are restricted to be non-negative, the scalars  $V_1^{ij}$  and  $V_2^{ij}$  must be non-negative. The conditions for  $L_o(j\omega_k) \in R_k \cup S_k$  then follow immediately from the formulation of the QFT constraints given in the previous section.  $\square$

Theorem 5.4 shows that fixing the phase of the PID controller between  $-90^\circ$  and  $90^\circ$  at two distinct frequencies, fixes the phase of the controller at every frequency. The Nyquist plot of the PID controller (a vertical straight line with real part  $k_p$ ) shows geometrically that in this case the three controller gains are uniquely determined (up to scaling) provided that  $-90^\circ < \psi_i < \psi_j < 90^\circ$  for  $\omega_i < \omega_j$ .

If the pure-derivative term in the controller is considered to be undesirable, the controller can be modified to the form:

$$K'(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1 + s\tau}$$



where  $\tau$  is a (fixed) sufficiently small parameter. In this case, Theorem 1 can be applied with minor modifications by redefining the uncertain plant as

$$G'(p, s) = \frac{G(p, s)}{1 + s\tau}$$

and solving for the new variables  $k_p' = k_p + k_i\tau$ ,  $k_i' = k_i$  and  $k_d' = k_d + k_p\tau$ . See [63] for details.

Using essentially the same arguments we can obtain a parallel result for the  $PDD^2$  (proportional-derivative-double derivative) controller  $K(s) = k_1 + k_2s + k_3s^2$ . Of course, this type of controller can be used only if the relative degree of the plant is at least two. Alternatively, parts of the feedback controller (e.g. integrators arising from steady-state error specifications) can be absorbed into the uncertain plant family, and the optimization can be carried out for the remaining (unspecified) part of the controller.

**Theorem 5.5:** (i) Let  $K(s) = k_1 + k_2s + k_3s^2$  with  $k_1$ ,  $k_2$  and  $k_3$  real parameters. Suppose that we impose the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  where  $\omega_i \neq \omega_j$  and  $\omega_i \tan \psi_j \neq \omega_j \tan \psi_i$ . Then all controllers of this form are fixed up to a scaling parameter  $\lambda \in \mathcal{R}$  and are parametrised as:

$$k_1 = \lambda \frac{\omega_i \omega_j (\omega_i \tan \psi_i - \omega_j \tan \psi_j)}{\omega_j \tan \psi_i - \omega_i \tan \psi_j}, \quad k_2 = \lambda \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_j \tan \psi_i - \omega_i \tan \psi_j}, \quad k_3 = \lambda$$

The magnitude and gain of  $K(s)$  at any frequency  $\omega$  is given as:

$$|K(j\omega)| = |\lambda| \frac{\sqrt{[(\omega_i^2 - \omega^2)\omega_j \tan \psi_i + (\omega^2 - \omega_j^2)\omega_i \tan \psi_j]^2 + \omega^2(\omega_i^2 - \omega_j^2) \tan^2 \psi_i \tan^2 \psi_j}}{|\omega_j \tan \psi_i - \omega_i \tan \psi_j|}$$

$$:= |\lambda| C_{ij}(\omega)$$

and

$$\arg K(j\omega) = \arctan \left( \frac{\omega(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{(\omega_i^2 - \omega^2)\omega_j \tan \psi_i + (\omega^2 - \omega_j^2)\omega_i \tan \psi_j} \right)$$

respectively.

(ii) If the controller gains  $k_1$ ,  $k_2$  and  $k_3$  are restricted to be non-negative, then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  with  $0 < \psi_i < \pi$  and  $0 < \psi_j < \pi$  are feasible if and only if

$$\frac{\omega_i \tan \psi_i - \omega_j \tan \psi_j}{\omega_j \tan \psi_i - \omega_i \tan \psi_j} \geq 0 \quad \text{and} \quad \frac{(\omega_i - \omega_j) \tan \psi_i \tan \psi_j}{\omega_j \tan \psi_i - \omega_i \tan \psi_j} \geq 0$$

In this case,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{\omega_k(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{(\omega_i^2 - \omega_k^2)\omega_j \tan \psi_i + (\omega_k^2 - \omega_j^2)\omega_i \tan \psi_j} \right)$$

Also,  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $|\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \notin [\phi_l \phi_h]$ .
- $|\lambda| \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)} \geq |\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  or  $|\lambda| \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C_{ij}(\omega_k)$  is defined in part (i).

**Proof:** Follows similarly to the proof of Theorem 5.3. Similar conclusions can also be drawn about the gain and phase of the open-loop system and its permissible regions subject to QFT constraints (details are omitted).  $\square$

### Phase lead/lag controller

We next consider the case of a first-order phase-lead (phase-advance) controller. These are widely used in classical control designs to improve the stability margins of the system, of to meet steady-state error specifications. The dual result for a phase-lag controller also follows easily.

**Theorem 5.6:** Let  $K(s) = k(s + b)/(s + a)$  with  $a > b > 0$  (“phase lead” controller). Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $0 < \psi_i < 90^\circ$  and  $0 < \psi_j < 90^\circ$  are feasible if and only if the following two conditions are satisfied:

$$\lambda := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} > 0 \text{ and } c := \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

In this case, the quadratic equation:

$$b^2 + \lambda b + c = 0$$

has one positive root  $b_+$  and one negative root  $b_-$  and the controller parameters  $b$  and  $a$  are determined uniquely as  $b = b_+$  and  $a = b_+ + \lambda = -b_-$ . In addition,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{\omega_k}{b} \right) - \arctan \left( \frac{\omega_k}{a} \right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l \phi_h]$ .
- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = \sqrt{(b^2 + \omega^2)/(a^2 + \omega^2)}$ .

**Proof:** The frequency response of the phase-lead controller is given as

$$K(j\omega) = k \frac{j\omega + b}{j\omega + a}$$

with gain and phase

$$|K(j\omega)| = k \sqrt{\frac{\omega^2 + b^2}{\omega^2 + a^2}} \quad \text{and} \quad \arg K(j\omega) = \arctan\left(\frac{\omega}{b}\right) - \arctan\left(\frac{\omega}{a}\right)$$

respectively. Now suppose we fix  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two frequencies  $\omega_i \neq \omega_j$ . Then,

$$\begin{aligned} \arctan\left(\frac{\omega_i}{b}\right) - \arctan\left(\frac{\omega_i}{a}\right) &= \psi_i \\ \arctan\left(\frac{\omega_j}{b}\right) - \arctan\left(\frac{\omega_j}{a}\right) &= \psi_j \end{aligned}$$

Using the trigonometric identity:

$$\tan(\alpha - \beta) = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta}$$

we get (after some algebra),

$$\omega_i^2 \tan \psi_i - \omega_i a + \omega_i b + ab \tan \psi_i = 0 \quad (5.19)$$

$$\omega_j^2 \tan \psi_j - \omega_j a + \omega_j b + ab \tan \psi_j = 0 \quad (5.20)$$

Multiplying equation 5.19 by  $\tan \psi_j$ , equation 5.20 by  $\tan \psi_i$  and subtracting the resulting two equations gives

$$a = b + \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} := b + \lambda$$

Since  $a > b$  for a phase-lead controller, we must have  $\lambda > 0$ . Substituting for  $a = b + \lambda$  in (1) leads to the quadratic equation:

$$b^2 + \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} b + \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} = 0$$

This must have a positive root  $b_+$  if the constraints are feasible, so that  $a = b_+ + \lambda > b_+ = b > 0$ . To see that *at most* one of the two roots of the quadratic equation

$$b = \frac{-\lambda \pm \sqrt{\lambda^2 - 4c}}{2}$$

is positive, note that the transfer functions  $(s + b)/(s + a)$  and  $(s - a)/(s - b)$  have identical phase responses; hence if one root of the quadratic is  $b$ , the other root must be  $-a$ . Formally, when  $\lambda > 0$  the roots of the quadratic can be classified as follows:

- $c < 0$ : One positive ( $b_+$ ) and one negative ( $b_-$ ) root.
- $c = 0$ : Zero and negative ( $b = -\lambda$ ) roots.
- $0 < c \leq \lambda^2/4$ : Here  $\sqrt{\lambda^2 - 4c} < \lambda$  so both roots are negative.
- $c > \lambda^2/4$ : Complex conjugate roots.

and so parameters  $a$  and  $b$  with  $a > b > 0$  are uniquely determined from the two phase conditions when  $\lambda > 0$  and  $c < 0$ . To show that  $b_- = -a$  note that

$$a = b_+ + \lambda = \frac{-\lambda + \sqrt{\lambda^2 - 4c}}{2} + \lambda = \frac{\lambda + \sqrt{\lambda^2 - 4c}}{2} = -b_-$$

The phase equation for  $L_o(j\omega_k)$  is immediate, while the gain inequalities on  $k$  for  $L_o(j\omega_k) \in R_k \cup S_k$  follow directly from the discussion of the previous section on noting that  $|K(j\omega_k)| = kC(\omega_k)$ . □

**Example:** Consider the following cases:

- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = 10^\circ$  and  $\psi_j = 30^\circ$ : We obtain  $\lambda = 11.9339$ ,  $c = -66.6806$  and so the constraints are feasible. The quadratic equation gives  $b = b_+ = 4.1467$  and  $a = b_+ + \lambda = -b_- = 16.0806$ .
- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = 60^\circ$  and  $\psi_j = 10^\circ$ : We obtain  $\lambda = 0.6785$ ,  $c = 0.6083$  and the roots of the quadratic are complex:  $b_{1,2} = -0.3393 \pm j0.7023$ . The constraints are infeasible.
- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = -10^\circ$  and  $\psi_j = 30^\circ$ : Clearly the constraints are infeasible for a lead (or a lag) controller. We obtain  $\lambda = 1.1905$ ,  $c = 7.7518$  and the quadratic has complex roots  $b_{1,2} = -0.5953 \pm j2.7198$ .

- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = -10^\circ$  and  $\psi_j = -30^\circ$ : Clearly constraints are infeasible for phase-lead controller (but not for a phase-lag controller). We obtain  $\lambda = -11.9339$ ,  $c = -66.6806$  while the quadratic equation gives  $b_+ = 16.0806$  and  $b_- = -4.1467$ .

The corresponding result for a phase-lag controller is as follows:

**Theorem 5.7:** Let  $K(s) = k(s + b)/(s + a)$  with  $b > a > 0$  (“phase lag” controller). Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $-90^\circ < \psi_i < 0^\circ$  and  $-90^\circ < \psi_j < 0^\circ$  are feasible if and only if the following two conditions are satisfied:

$$\lambda := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0 \quad \text{and} \quad c := \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

In this case, the quadratic equation:

$$b^2 + \lambda b + c = 0$$

has one positive root  $b_+$  and one negative root  $b_-$  and the controller parameters  $b$  and  $a$  are determined uniquely as  $b = b_+$  and  $a = b_+ + \lambda = -b_-$ . In addition,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan\left(\frac{\omega_k}{b}\right) - \arctan\left(\frac{\omega_k}{a}\right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l, \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = \sqrt{(b^2 + \omega^2)/(a^2 + \omega^2)}$ .

**Proof:** Along similar lines to the proof of Theorem 5.6. □

Theorems 5.6 and 5.7 show that fixing the phases of the phase lead or phase lag controller in the intervals  $(0^\circ, 90^\circ)$  or  $(-90^\circ, 0^\circ)$  respectively determines uniquely the dynamic part of the controller when the constraints are feasible. Feasibility of the constraints is easily checked from two sign conditions, and the controller parameters are determined by solving a quadratic equation.

## Second-order controller with complex poles or zeros

We finally consider the case of a second-order controller with complex (conjugate) poles. The corresponding dual result for second-order controllers with complex zeros also follows easily.

**Theorem 5.8:** (i) Let  $K(s) = k/(s^2 + 2\zeta\omega_n s + \omega_n^2)$  with  $\omega_n > 0$  and  $0 < \zeta < 1$  (“complex-pole second-order lag”). Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $-180^\circ < \psi_i < 0^\circ$  and  $-180^\circ < \psi_j < 0^\circ$  and  $\psi_i \neq \psi_j$  are feasible if and only if the following conditions are satisfied:

- If  $\psi_i \neq -90^\circ$  and  $\psi_j \neq -90^\circ$ ,

$$\lambda = \frac{\omega_j \tan \psi_j - \omega_i \tan \psi_i}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} > 0$$

and

$$0 < \zeta := \frac{\tan \psi_i}{2} \left( \sqrt{\frac{\omega_j \lambda}{\omega_i}} - \sqrt{\frac{\omega_i}{\lambda \omega_j}} \right) < 1 \quad (5.21)$$

in which case  $\omega_n$  and  $\zeta$  are defined uniquely as  $\omega_n = \sqrt{\omega_i \omega_j \lambda}$  and via (3) respectively.

- If  $\psi_j = -90^\circ$  then either of the two conditions must hold: (i)  $\psi_i \in (-90^\circ, 0^\circ)$  and  $\omega_j < \omega_i$ , or (ii)  $\psi_i \in (-180^\circ, -90^\circ)$  and  $\omega_j > \omega_i$ , in addition to the condition

$$\zeta := \frac{(\omega_j^2 - \omega_i^2) \tan \psi_i}{2\omega_i \omega_j} < 1 \quad (5.22)$$

in which case  $\omega_n$  and  $\zeta$  are uniquely determined as  $\omega_n = \omega_j$  and via (4), respectively.

- If  $\psi_i = -90^\circ$  then either of the two conditions must hold: (i)  $\psi_j \in (-90^\circ, 0^\circ)$  and  $\omega_i < \omega_j$ , or (ii)  $\psi_j \in (-180^\circ, -90^\circ)$  and  $\omega_i > \omega_j$ , in addition to the condition

$$\zeta := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_j}{2\omega_i \omega_j} < 1 \quad (5.23)$$

in which case  $\omega_n$  and  $\zeta$  are uniquely determined as  $\omega_n = \omega_i$  and via (5), respectively.

(ii) When the phase conditions are feasible we have

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{2\zeta\omega_n\omega_k}{\omega_n^2 - \omega_k^2} \right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = 1/\sqrt{(\omega_n - \omega)^2 + 4\zeta^2\omega_n^2\omega^2}$ .

**Proof:** The frequency response of the controller is given by

$$K(j\omega) = \frac{k}{\omega_n^2 - \omega^2 + 2j\zeta\omega_n\omega}$$

from which its magnitude and phase responses can be obtained as:

$$|K(j\omega)| = \frac{k}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} = kC(\omega) \text{ and } \arg K(j\omega) = \arctan \left( \frac{2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \right)$$

respectively. Setting  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  gives

$$\frac{2\zeta\omega_n\omega_i}{\omega_n^2 - \omega_i^2} = \tan \psi_i \text{ and } \frac{2\zeta\omega_n\omega_j}{\omega_n^2 - \omega_j^2} = \tan \psi_j \quad (5.24)$$

for  $\psi_i \neq -90^\circ$  and  $\psi_j \neq -90^\circ$ . Solving simultaneously the above two equations gives:

$$\omega_n^2 = \frac{\omega_i\omega_j(\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i}$$

which defines  $\omega_n$  uniquely iff  $\lambda > 0$ . Substituting into (6) then gives the expression for  $\zeta$  and the corresponding condition for an under-damped response ( $0 < \zeta < 1$ ). When  $\psi_j = -90^\circ$  we must have  $\omega_n = \omega_j$  and hence  $\zeta$  is given by (5). This is positive when  $(\omega_j - \omega_i) \tan \psi_i > 0$  from which the two stated conditions follow. Finally, the phase equation for  $L_o(j\omega_k)$  follows immediately, while the gain conditions for  $L_o(j\omega_k) \in R_k \cup S_k$  can be derived from the discussion in the previous section on QFT constraints.  $\square$

Again, Theorem 5.9 shows that fixing the phase of the controller at two distinct frequencies, determines completely the dynamic part of the controller, when the

constraints are feasible. In the Theorem formulation the controller is restricted to be under-damped. This restriction can be removed, if required, by ignoring throughout the  $\zeta < 1$  condition. An almost identical procedure may be used to determine the dynamic part of a minimum-phase non-proper controller  $k(s^2 + 2\zeta\omega_n s + \omega_n^2)$  from its two phases in the interval  $(0^\circ, 180^\circ)$  at two distinct frequencies  $\omega_i$  and  $\omega_j$ ; details are omitted.

In all three cases considered above we have derived simple gain conditions on the nominal open-loop gain, so that the QFT robust stability and performance constraints are satisfied. These are of the form  $L_o(j\omega_k) \in R_k \cup S_k$ , which for a fixed phase  $\arg L_o(j\omega_k) = \phi_k$ , correspond to gain intervals

$$[k_1(\omega_i, \phi_k), k_2(\omega_i, \phi_k)] \cup [k_3(\omega_i, \phi_k), \infty]$$

where  $i = 1, 2, \dots, N$ . Thus the optimisation problem takes the form: “Minimise the optimisation criterion (i.e. cross-over frequency, closed-loop bandwidth, asymptotic open-loop gain, etc), so that for each design frequency  $\omega_i$ ,  $i = 1, 2, \dots, N$ ,  $L_o(j\omega_k) \in S_k \cup R_k$  and Nyquist’s encirclement criterion is satisfied”. Since for each of the three types of controllers described above the phase of the nominal open-loop system is completely determined once two controller phases have been fixed, we can use the following algorithm for solving the optimisation problem:

#### 5.4.5 Optimisation algorithm steps:

1. Obtain a phase array  $\Phi$  by discretising the phase interval  $(-360^\circ \ 0^\circ]$ .
2. Select any two distinct frequencies  $\omega_i$  and  $\omega_j$  in the set of design frequencies  $(\omega_1, \omega_2, \dots, \omega_N)$ .
3. Calculate the phase intervals  $\Phi_k \subseteq \Phi$  and  $\Phi_l \subseteq \Phi$  in which the nominal open loop phase  $\arg L_o(j\omega)$  can vary at  $\omega = \omega_k$  and  $\omega = \omega_l$  respectively. These depend on the type of controller to be designed, e.g. for a PID controller they lie within  $\pm 90^\circ$  of  $\arg G(p_o, \omega_k)$  and  $\arg G(p_o, \omega_l)$ , etc.
4. Initialise an  $m \times n$  array  $F$  where  $m$  and  $n$  are the sizes of  $\Phi_k$  and  $\Phi_l$  respectively, to contain the value of the objective function (cross-over frequency, closed-loop bandwidth, asymptotic gain etc.) for each phase pair. Also, initialise  $m \times n$



controller parameter arrays to contain the parameters, e.g.  $(k_p, k_d, k_i)$  for a PID controller,  $(k_o, a, b)$  for a phase-lead/lag controller, or  $(k_o, \omega_n, \zeta)$  for a second-order controller with complex poles/zeros.

5. For each  $(\Phi_k(i), \Phi_l(j)) \in \Phi_k \times \Phi_l \subseteq \mathcal{R}^{m \times n}$ :

- (a) Calculate  $\psi_i = \Phi_k(i) - \arg G(p_o, \omega_k)$  and  $\psi_j = \Phi_l(j) - \arg G(p_o, \omega_l)$ .
- (b) Determine a controller  $K(s)$  of one of the above types, such that  $\arg K(j\omega_k) = \psi_i$  and  $\arg K(j\omega_l) = \psi_j$ . If these phase constraints are infeasible, set  $F(i, j) = \infty$  and consider the next phase pair  $(\Phi_k(i), \Phi_l(j))$ .
- (c) Find the minimum value of gain  $k_o > 0$  such that (i)  $L_o(j\omega_q) = k_o K(j\omega_q) G(p_o, j\omega_q) \in R_q \cup S_q$  for all  $q = 1, 2, \dots, N$  and (ii)  $L_o(j\omega)$  satisfies Nyquist's encirclement criterion. If no such gain  $k_o$  exists, set  $F(i, j) = \infty$  and consider the next phase pair  $(\Phi_k(i), \Phi_l(j))$ .
- (d) Calculate the value of the objective function (cross-over frequency, closed-loop bandwidth, asymptotic gain, etc) corresponding to the designed  $L_o(j\omega)$  and assign it to the  $(i, j)$ th element of  $F$ . Save also the controller parameters to the corresponding entries of the parameter arrays.

6. At the end of all  $m \times n$  iterations, calculate  $\gamma_o = \min_{(i,j) \in \Phi_k \times \Phi_l} (F)$  and  $(i^*, j^*) \in \operatorname{argmin}(F)$ . If  $\gamma_o = \infty$  the QFT constraints are infeasible; otherwise the optimal cost is  $\gamma_o$  and the optimal controller parameters can be obtained from the  $(i^*, j^*)$ th elements of the controller-parameter arrays.

A few remarks on the algorithm:

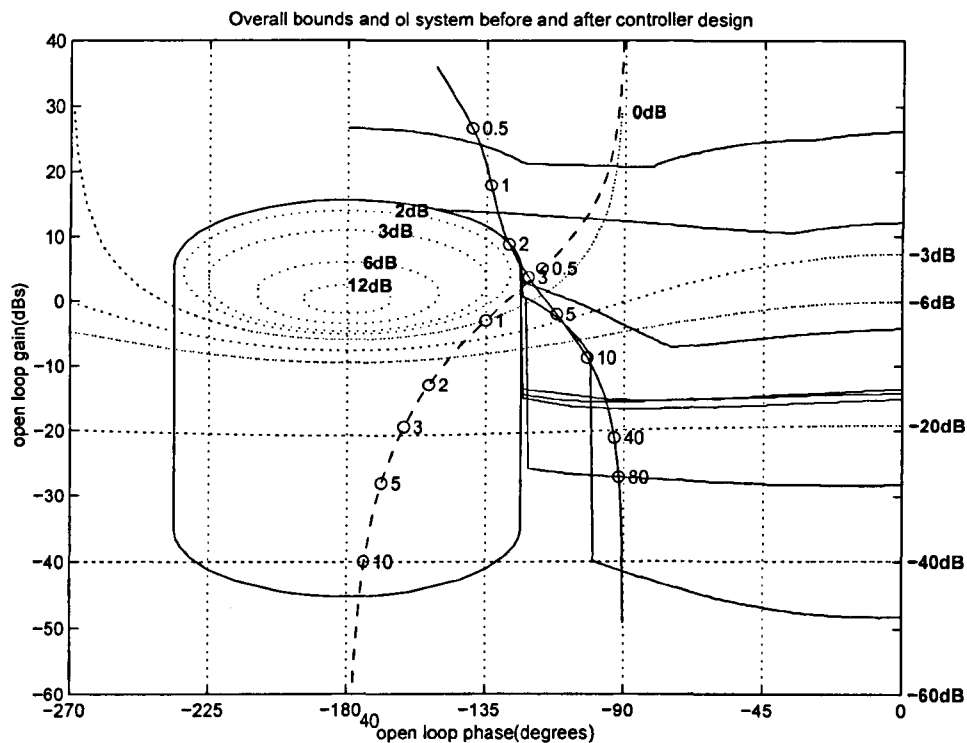
- In step (1) of the algorithm the phase discretisation of the interval  $(-360^\circ, 0^\circ]$  results in a phase grid  $\Phi$ , typically equally spaced. In practice, 50–100 phases are adequate. It is helpful to calculate the performance bounds (“Horowitz-tracking”, “Horowitz-disturbance-rejection” templates and  $\mathcal{U}$ -contour) over the same phase grid.
- In principle any two frequencies  $\omega_k$  and  $\omega_l$  can be selected from the set of design frequencies in step (2). In general, selecting these frequencies reasonably far-apart (for minimum numerical sensitivity) works well in practice. A common-

sense rule is to choose frequencies at which the controller can introduce a wide-range of phase without conflicting with the QFT constraints or the expected characteristics of the system; for example, if the nominal plant is of type zero and the controller introduces integral action, the open-loop phase at very low frequencies will be near  $-90$  degrees, and therefore frequencies in this range should not be selected.

- In steps (3) and (5a) of the algorithm all phase calculations can be performed modulo  $-360^\circ$ . This restricts the phase interval of interest to the range  $(-360^\circ, 0^\circ]$ .
- Since the phase of  $L_o(j\omega)$  is completely determined when two controller phases are fixed, the calculation of the minimum gain in step (5c) is straightforward. For example, one possible method is to calculate the minimum distance between the plant and the corresponding “open” performance bounds and check whether this amount of gain brings the high design frequencies within the  $\mathcal{U}$ -contour, together with a stability test. Checking the total number of encirclements required for stability is also straightforward and can be performed by purely graphical means (i.e. by counting the crossings of the  $-180^\circ$  line and their directions). See [13], [12] for details. Note also, that a frequency grid “denser” than the set of design frequencies must typically be used for this purpose.
- Step 5(c) requires the calculations of the performance bounds at arbitrary phases, which may not coincide with the discretised phases of array  $\Phi$ . There is no difficulty, however, in estimating the performance gains from adjacent phase points, e.g. using linear interpolation. Alternatively, the performance bounds may be calculated exactly at these phases to arbitrary accuracy using a bisection algorithm implemented between steps (5b) and (5c). In practice, however, it is sufficient to substitute each point with the one which is closest on the pre-defined phase-grid.

#### 5.4.6 Design example

To demonstrate the procedure, the uncertain system defined in the previous chapter is used here to demonstrate the optimal loop-shaping method for a PID controller. The phase grid was discretised in the interval  $[-180^\circ, 0]$  using 180 equally spaced



points. Using the optimization procedure described in the previous section, the optimal controller gains were found as:

$$k_p = 9.360 \quad k_i = 6.473 \quad \text{and} \quad k_d = 5.290$$

Figure 5.5 shows the nominal plant frequency response, together with the optimised open-loop response in the Nichol's chart. The 8 design frequencies are marked with circles. Note that the optimal design satisfies the robust stability and performance constraints, in the sense that it lies on or above the Horowitz templates and does not penetrate the  $U$ -contour. All points are reasonably close to the templates, which indicates that no excessive gains have been used. Figure 5.6 shows that closed-loop frequency responses after the design of a pre-filter. As required, they all lie within the two specified bounds which are also displayed. Finally, a number of step responses corresponding to a wide range of uncertain parameters are shown in Figure 5.7.

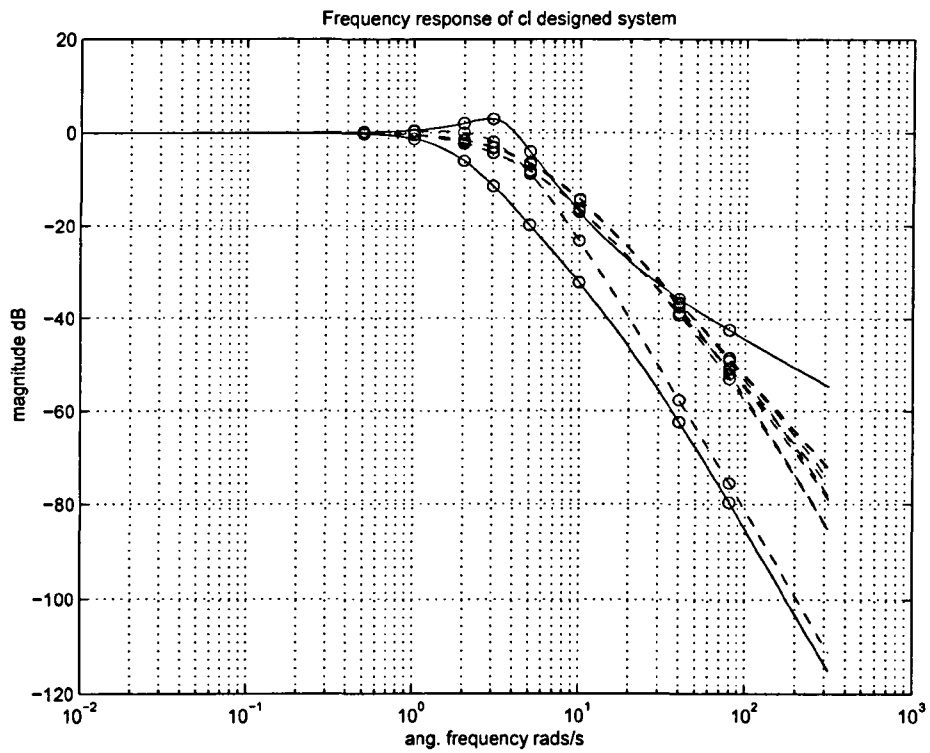


Figure 5.8: *Closed-loop system frequency response*

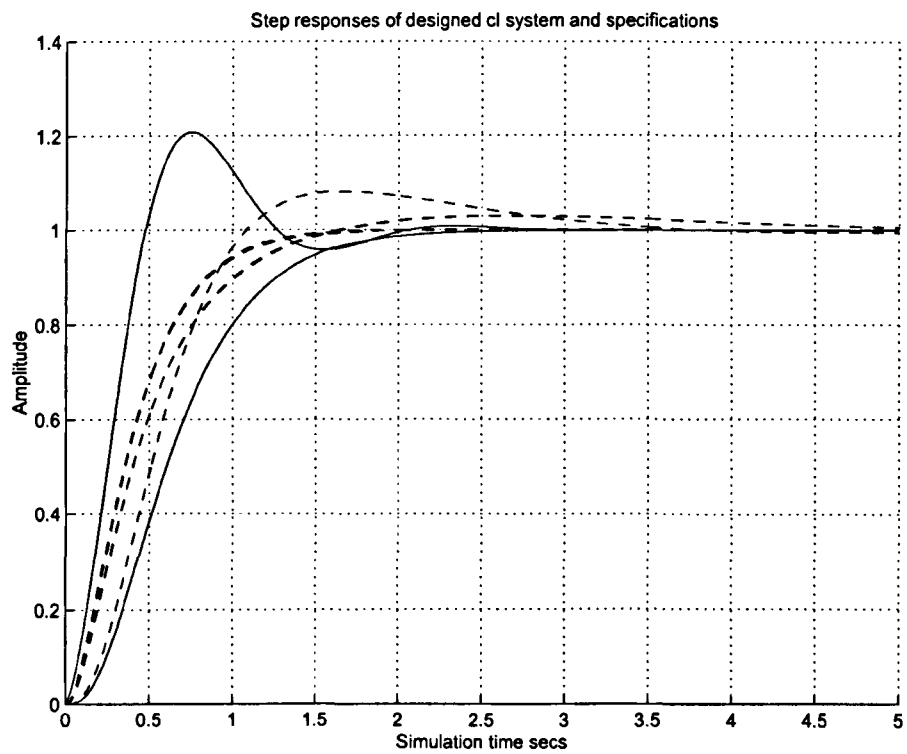


Figure 5.9: *Step responses of plants*

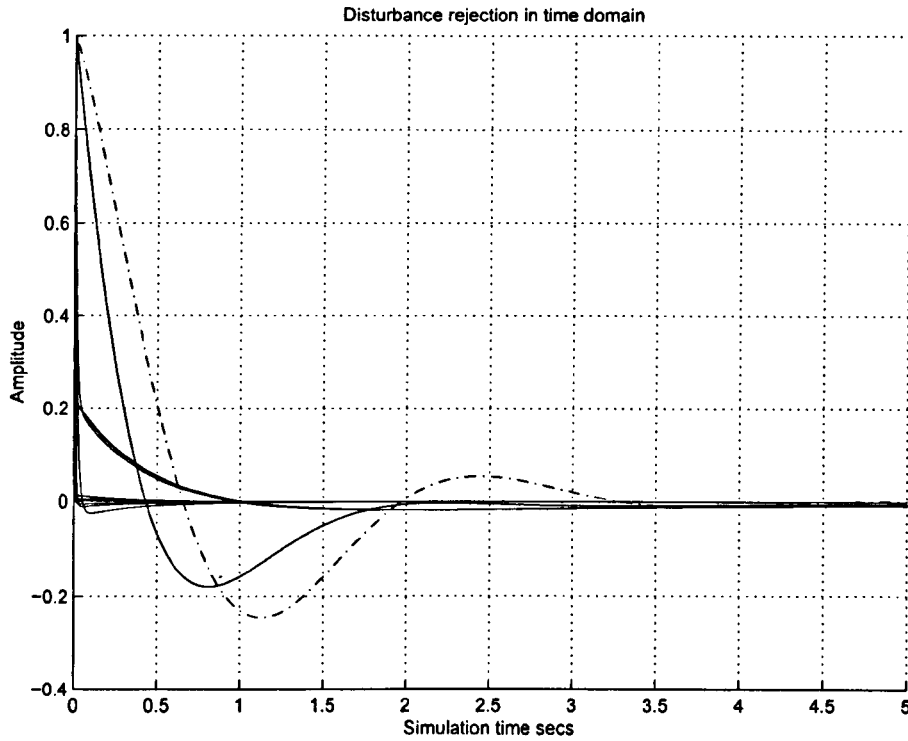


Figure 5.10: *Disturbance rejection in the time domain*

## 5.5 Conclusions

Two general algorithms for automating the loop-shaping procedure of the QFT design have been presented in this chapter. The first algorithm formulates the design problem as a Linear Programme which can be solved subject to robust stability and performance constraints derivable from the Horowitz and  $U$ -contours, together with analyticity and realisability constraints obtained by descretising Bode's gain-phase integral relation.

The second algorithm optimises the parameters of fixed-structure compensators. The algorithm is simple, easy to implement, and can be used to automate the loop-shaping step of the QFT design procedure. It can be used to design robust-performance optimal controllers of a simple structure (PID, phase-lead/lag, second-order), or more complex controllers involving arbitrary interconnections of these structures. The design algorithm has been illustrated with a simple design example. Extensions of the method to multivariable systems are possible using the standard QFT approach [57].

# Chapter 6

## QFT Software Toolbox

### 6.1 Introduction

This chapter describes in details a Matlab-based toolbox developed to implement the *QFT* design procedure described in previous chapters. This is a library of Matlab functions and script files which perform all tasks required in the design procedure. The main functions can be divided in the following categories:

- Graphical functions related to the Nichol's chart environment and its associated templates ( $M$  and  $N$  circles) and design simulation results (frequency-domain tracking bounds, step response simulations for specified combinations of uncertain parameters, etc).
- Computational and graphical functions specific to the QFT design environment, e.g functions to calculate and display the plant's uncertainty templates (including their simplification via a convex hull algorithm), robust stability and performance contours ( $U$ -contour, Horowitz templates), etc.
- A graphical loop-shaping tool for designing feedback controllers and pre-filters.
- Optimization based routines for designing feedback controllers and pre-filters.
- Other utility tools related to the overall design, e.g. functions to determine closed-loop stability/instability properties, functions for discretising the Bode integral gain-phase relation and for setting up linear constraints, etc.

All routines are implemented in Matlab's environment and use extensively many of the built-in computational and graphical resources for performing intermediate calcu-

lations, graphical representations and data handling. The overall aim of the work is to create a user-friendly environment for carrying out robust QFT designs of at least medium complexity. The toolbox is designed to have an open architecture and can easily incorporate future tools and design techniques in the QFT area.

In the following section the main computational and graphical functions of the toolbox are described in detail.

## 6.2 Template Generation

*Uncertainty templates* represent the uncertainty of the plant in the Nichol's chart. As explained in earlier chapters, there are two main sources of model uncertainty: (i) Parametric uncertainty, e.g. from incomplete knowledge of the parameters describing the system's differential equations, state-space model or transfer function, and (ii) unstructured uncertainty, reflecting imperfect information about the model structure, actuator and sensor dynamics, or high frequency dynamics which are sometimes purposefully ignored when setting up the model of the plant. Of these two sources, parametric uncertainty is more difficult to handle in control design in terms of the system's robust stability and performance properties.

The software concentrates on parametric uncertainty (although unstructured uncertainty models can also be easily incorporated). Uncertain parameters are assumed to vary independently between known upper and lower bounds. Each of these intervals is gridded uniformly using  $n$  points. Thus if there are  $m$  uncertain parameters the  $m$ -dimensional hypercube in parameter space is represented by  $n^m$  discrete points. For each design frequency, the frequency response of the plant is calculated at each of these  $n^m$  points and its magnitude and phase is calculated, which then define the co-ordinates of a point in the Nichol's chart. The collection of all these  $n^m$  points on the Nichol's chart defines an uncertainty template at the corresponding design frequency.

Although the method appears to be rather crude (and more efficient template generation techniques have been proposed [4], [22]) it is easy to implement and works well in practice (say up to 10 uncertain parameters using 4 discretisation points) which

is sufficient for most applications. Note that uncertainty templates are generated only once at the beginning of each design. In practice the uncertainty information included in each uncertainty template using this method is highly redundant, as only the extreme points lying on the template's boundary are relevant to subsequent calculations (e.g. the construction of the Horowitz templates or the  $U$  contour). Thus, in order to speed up the design, the convex hull of the  $m^n$  points defining the template is calculated (at each frequency) and used to represent the uncertainty template. This increases dramatically the execution time of all subsequent design steps with only minimal amount of conservativeness introduced in the design.

### 6.2.1 Algorithm Pseudo-Code

The design algorithm is described here for the simple system with two uncertainty parameters  $k$  and  $a$  introduced in earlier chapters of the thesis, and  $n$  design frequencies. The plant has the form:

$$G(s) = \frac{ka}{s(s+a)}$$

where both  $a$  and  $k$  are assumed to be uncertain. The procedure is as follows

- Initialize the required variables.
- Calculate the magnitude and phase of the plant at every design frequency by plotting the Bode response of the system.
- Store the data in the form of an array of  $n$  rows and  $n_a n_k$  columns, corresponding to the number of discretised points of the  $a$  and  $k$  parameter, respectively.

A set of uncertainty template generated using the matlab code provided in Appendix C section 9.3.2 corresponding to the previous example at the specified 8 design frequencies is shown in figure 6.1.

## 6.3 Convex Hull

The output of the  $M$ -file for creating the uncertainty templates contains points which are spread out in the Nichol's chart. The points in the template which are of primary use are the boundary points. The process of forming the smallest convex polygon that includes all points is called the convex hull of the point set. There are various methods



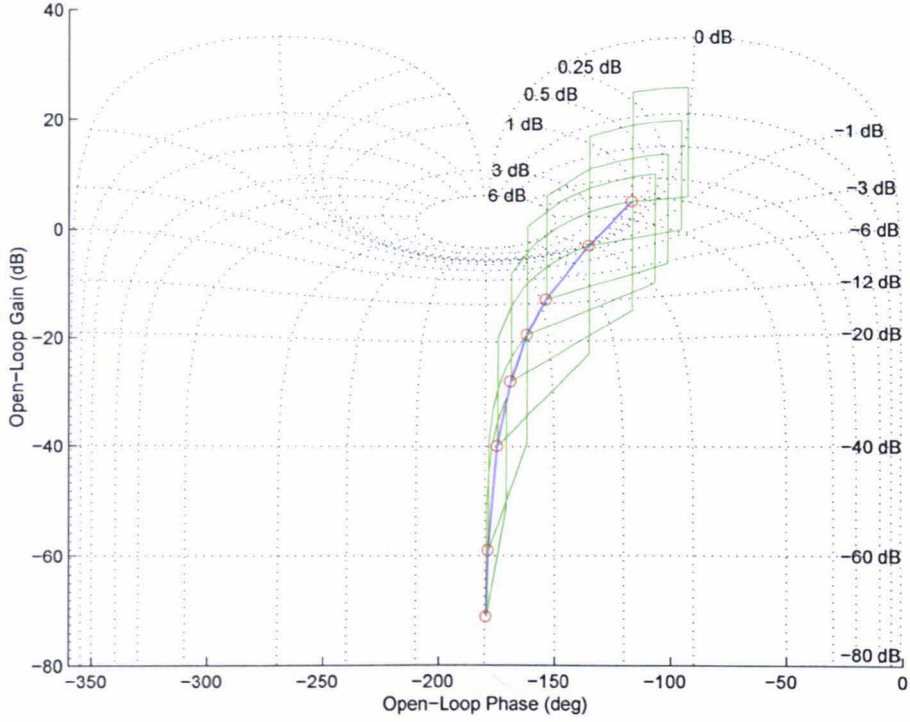


Figure 6.1: *Nominal Open-loop and Uncertainty Templates*

to compute the convex hull of a number of points, e.g. see [Ref], [Ref]. The convex hull generation method used in this toolbox is outlined next.

Let  $n$  be the number of uncertainty template points on the plane. The convex hull construction proceeds by ordering the points according to their phase relative to a point interior to the convex hull. First, identical points are eliminated from the array and three distinct points on the boundary of the convex hull are selected by selecting (i) two points with a maximum and minimum  $x$ -coordinate, respectively, and (ii) a point with minimum  $y$ -coordinate (if more than one point are extreme with respect to a particular co-ordinate any one will do). The three points selected this way will certainly lie on the boundary of the hull, and thus their centroid will be an interior point of the hull. Re-defining temporarily this point as the origin by parallel translation of the  $x$  and  $y$  axes, all points in the array are next rearranged in ascending phase (relative to the re-defined origin).

The algorithm proceeds by eliminating points in the interior of the hull. Starting from the first point of the array (arbitrarily a point of minimum  $y$ -coordinate), three

consecutive points (of increasing phase), say  $A, B$  and  $C$  are tested in turn, moving in the counter-clockwise direction. A subroutine determines whether the three points define a “left turn”, i.e. whether point  $C$  lies to the left of the directed line  $AB$ . As long as the answer is affirmative, point  $B$  “passes the test” and remains (for the time being) in the array; an integer counter is incremented and the next three points are considered (say  $B, C$  and  $D$ ). If the answer is negative, point  $B$  cannot possibly lie on the boundary of the convex hull and is eliminated from the array; the algorithm then backtracks by decrementing the counter to determine whether additional points in the array preceding  $B$  can be eliminated. The algorithm continues this way, alternating between forward and reverse sweeps until the last point is encountered, at which point the points on the array which have survived define the convex hull.

To illustrate the algorithm consider figure 6.2 and suppose that  $P_1$  is the point of minimum  $y$  co-ordinate. Assume also that points in the array have been ordered in increasing phase relative to an interior point  $C$  as shown in the figure. Start by setting the counter to 1 and considering the triplet  $(P_1, P_2, P_3)$ , i.e. test point  $P_2$  by determining whether  $P_3$  lies to the left of the directed line  $P_1P_2$ . In this case the answer is yes, so  $P_2$  passes the test, the counter is incremented to 2 and the point triplet  $(P_2, P_3, P_4)$  is next considered. Here  $P_3$  fails the test as  $P_4$  lies to the right of the directed line  $P_2P_3$ . Thus  $P_3$  is eliminated, the counter is decremented to 1 and the triplet  $(P_1, P_2, P_4)$  is next considered. Again  $P_2$  passes the test (for the second time), so the counter is incremented to 2 and the triplet  $(P_2, P_4, P_5)$  is considered. By continuing this way the convex hull is finally constructed [63].

### 6.3.1 Algorithm Pseudo-Code

The convex hull algorithm described above was implemented in Matlab.

- The input is an array of complex numbers which defines the uncertainty template of the plant at a design frequency.
- The output is the (ordered) convex hull of the input array of complex numbers.

Upon on the availability of input data, the algorithm is applied to all uncertainty templates defined at various frequencies. The design algorithm pseudo-code is as follows:

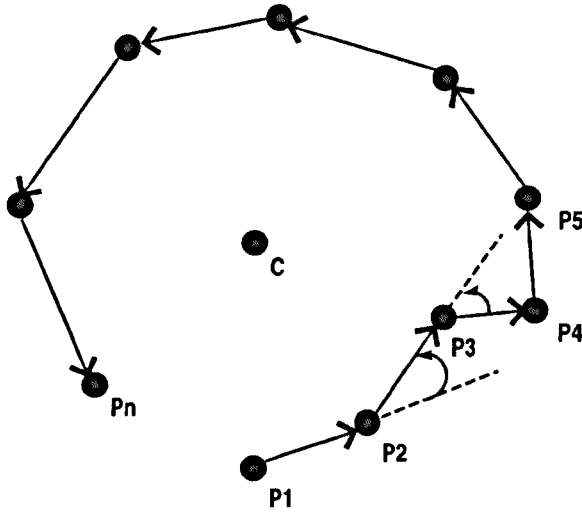


Figure 6.2: Convex hull design

- Analyse the input data and separate their imaginary and real parts.
- Remove identical points from the data.
- Find a point inside the hull. One such point is the centroid of three points known to lie in the hull, i.e.

$$C = \left( \frac{x_1 + x_2 + x_3}{3} \right) + i \left( \frac{y_1 + y_2 + y_3}{3} \right)$$

where the three points  $(x_i, y_i)$  have extreme (minimum or maximum) co-ordinates.

- Calculate the phase of input array relative to centroid.
- Find point of minimum  $y$ -coordinate and re-order array in counter-clockwise direction with it as the starting point.
- Based on the three-point test described above, and starting from the first point of the array, determine the extreme points of the convex hull.
- Output the convex hull as a complex array.

Figure 6.3 shows the convex hull of the uncertainty templates of the uncertain plant of the previous example generated using the matlab code provided in Appendix C section 9.3.4, defined at 6 design frequencies.

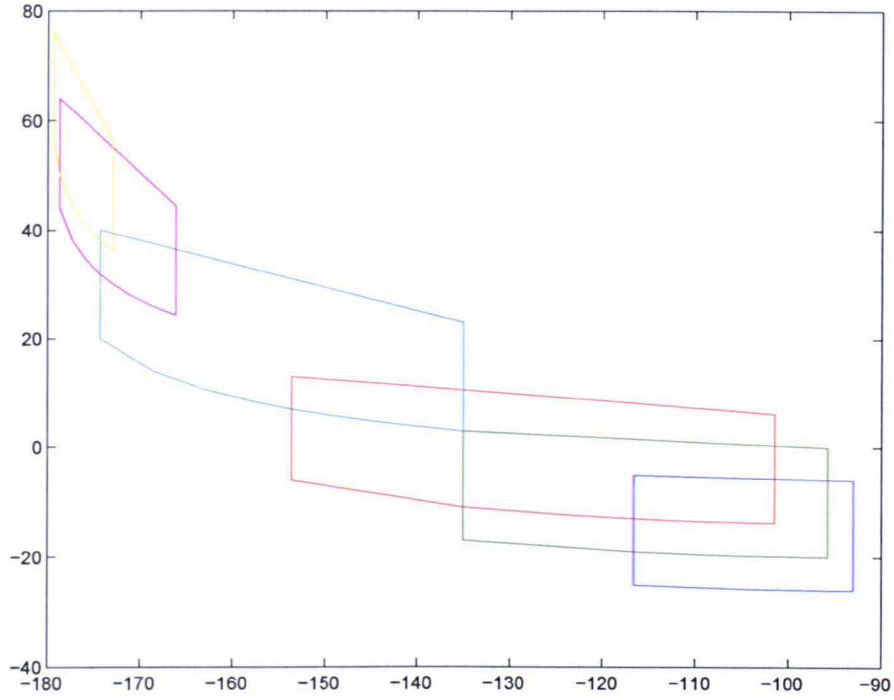


Figure 6.3: Convex hull of the uncertainty template at six frequencies

## 6.4 Horowitz Bounds (Tracking)

The tracking bounds define the minimum open-loop gain required to meet the closed-loop tracking specifications for every plant in the uncertainty set. The tracking bounds are designed based on the maximum allowable gain variation of the closed-loop specification  $\delta_R(\omega_i)$  at a frequency  $\omega_i$ . The allowable gain variation is defined by the user in the form of *Upper bound* and *Lower bound* of the closed-loop gain variation. Typical bounds are shown in Figure 6.1. The gain difference is defined as:

$$\delta_R(\omega_i) = |T_{RU}(j\omega_i)| - |T_{RL}(j\omega_i)|$$

where the gain is measured in *dB*.

### 6.4.1 Algorithm Pseudo-Code

The theory behind the definition of *Horowitz Bounds* was given in chapter 4. This section outlines an automatic design algorithm for generating the bounds at a specific design frequency.

The input data required for the design are:

- Uncertainty template (row vector of complex numbers) defined in the Nichol's chart. Real part describes the phase in degrees, imaginary part is gain in  $dB$ .
- Reference point of template corresponding to nominal plant (this could be an arbitrary fixed point inside template, first point in template etc.).
- Minimum and Maximum open-loop gain in  $dB$ ,  $L_{max}$  and  $L_{min}$ , respectively (used to specify the range of open-loop gain).
- Minimum open-loop phase in degrees (typically  $-360^\circ$ ).
- Maximum open-loop phase in degrees (typically  $0^\circ$ ).
- Number of phase points used to define phase grid (linearly spaced between minimum and maximum phase).
- Maximum allowable gain span (dB) that closed-loop responses must satisfy for all points of the uncertainty template.
- Magnitude tolerance (db) which specifies the required accuracy of the algorithm.

A reference point is chosen from the *Uncertainty template* to indicate the nominal plant. The choice of the nominal point is arbitrary, but it is usual to choose the point of the template corresponding to the minimum phase or gain. Upon the availability of the required input data the algorithm will:

- Check for data consistency.
- Discretise the phase range (maximum phase to minimum phase interval) to  $n$  points as specified by the user.
- Obtain magnitude and phase of the reference point.
- Start the iteration to obtain the *Horowitz gain* for every point of the phase-

grid; this involves the following steps:

- Place the uncertainty template on the vertical phase-line of the chart corresponding to the current grid phase point (say  $\phi$ ), i.e. re-define all phase coordinates of the template as  $\psi_i \leftarrow \psi_i - \psi_{ref} + \phi$  where  $\psi_i$  is the  $i$ -th phase point of the uncertainty template and  $\psi_{ref}$  the phase of the point of the template corresponding to the nominal plant; select initially the vertical position (gain) of the reference point as  $(L_{max} - L_{min})/2$  and redefine the gain of the remaining points on the template accordingly.
- Determine whether the maximum closed-loop gain difference ( $M$ -circle value)  $\Delta M_{max}$  among all points of the template in its current position matches the specified gain spread (within the specified tolerance).
- If the answer is affirmative, mark the gain as a point on the Horowitz template for the current phase (and design frequency), break the loop and consider the next phase on the grid. If  $\Delta M_{max}$  exceeds the specified tolerance more gain needs to be injected - hence position the template at the current phase-line at a vertical distance  $(3L_{max} + L_{min})/4$  from the origin and repeat the test; If  $\Delta M_{max}$  is less than the specified tolerance, the gain is too large and can be decreased - hence position the template at the current phase-line at a vertical distance  $(L_{max} + 3L_{min})/4$  from the origin and repeat the test. Repeat this bisection procedure until  $\Delta M_{max}$  matches the specifications (within the specified tolerance).
- Repeat the procedure for the next phase-grid point.
- Repeat the procedure for the next design frequency.

At the end of the algorithm the Horowitz templates have been calculated using the Matlab code provided in Appendix C section 9.3.5 and can be plotted on Nichol's chart.

## 6.5 Disturbance rejection bounds

The need for calculating disturbance rejection bounds was described in chapter 4. Here an automatic design algorithm for disturbance rejection is outlined. A second-order

disturbance-rejecting model has the form:

$$M_D(s) = \frac{Y(s)}{D(s)} = \frac{s(s+a)}{(s+a)^2 + b^2}$$

where

$$a = \frac{1}{t_x} \ln \left( \frac{\alpha_p}{D_o} \right) = \zeta \omega_n$$

$$b = \omega_n (1 - \zeta)^{-\frac{1}{2}}$$

Robust disturbance-rejection based on this model imposes constraints on the sensitivity function of the form

$$\left| \frac{1}{1 + K(j\omega_i)G(p, j\omega_i)} \right| \leq |M_D(j\omega_i)|$$

Combined with tracking bounds, disturbance-rejection bounds define the overall robust performance bounds. These are represented graphically in the form of Horowitz templates by calculating the pointwise maximum between the two corresponding templates (tracking and disturbance rejection) at every design frequency.

The algorithm is similar to the algorithm used for designing the Horowitz bounds.

### 6.5.1 Algorithm Pseudo-Code

The algorithm is similar to the algorithm used for constructing the contours for the robust tracking bounds. The main difference is that instead of calculating the gain spread at each position of the template, in this case the maximum gain of the sensitivity function needs to be calculated among all points in the template Matlab code provided in Appendix C section 9.3.6.

## 6.6 High Frequency Contour (*U*-Contour)

To enforce adequate phase margins on the design for all uncertain plants, the open-loop response of the system should not approach the critical point, or equivalently the closed loop responses should not exhibit large peaks in their frequency response. To enforce this design objective a minimum amount of damping is imposed on the system in the form of an *M*-circle of a certain value that the open-loop responses of all perturbed systems are not allowed to penetrate. This robust stability condition is enforced by ensuring the system response of the perturbed system at high frequencies (which under

the assumption of parametric uncertainty typically has the form of a vertical line) does not penetrate an appropriate region around the critical point. This region is defined by extending the lower part of the  $M$ -circle by an appropriate amount, specified by the gain spread of the high frequency uncertainty template. The value ( $V$ -dB) by which the  $M$ -circle is extended is given by:

$$\begin{aligned}\Delta \log P &= \lim_{\omega \rightarrow \infty} [\log |P_{max}(j\omega)| - \log |P_o(j\omega)|] \\ &= VdB\end{aligned}$$

where  $P_{max}(j\omega)$  denotes the frequency response of the plant defined at the parameter corresponding to maximum gain, while  $P_o(j\omega)$  is the plant's nominal frequency response (determined for an arbitrary parameter vector which should remain fixed over all design frequencies).

### 6.6.1 M-Circle

Before constructing the high frequency contour we first need to construct the  $M$ -circle in the Nichol's chart. Constraining the open-loop response of the nominal system not to penetrate the  $M$ -contour in the Nichol's chart means that the magnitude frequency response of the nominal complementary sensitivity function will be uniformly bounded from above by  $M$ . As shown in chapter 3, the equation of the  $M$ -circle in the Nyquist diagram is given by:

$$\left(u + \frac{M^2}{M^2 - 1}\right)^2 + v^2 = \left(\frac{M}{M^2 - 1}\right)^2$$

Here  $u$  denotes the real variable of the polar plot and  $v$  the imaginary variable. The above equation represents a circle with its centre at  $\left(-\frac{M^2}{M^2 - 1}, 0\right)$  and radius  $\left|\frac{M}{M^2 - 1}\right|$  in the polar (Nyquist) plane. The circle needs to be translated to a corresponding contour in the Nichols chart. The algorithm considers separately the cases  $M > 1$ ,  $M = 1$  and  $M < 1$ . The first case results in closed contours defined only for certain phases; the second case results in an open contour defined only for certain phases, while the third case results in open contours defined for every phase point. The algorithm is also used to construct an  $M$ -contour grid on the Nichols chart which is useful when loop shaping is carried out by the designer manually (i.e. without relying on an optimization algorithm).



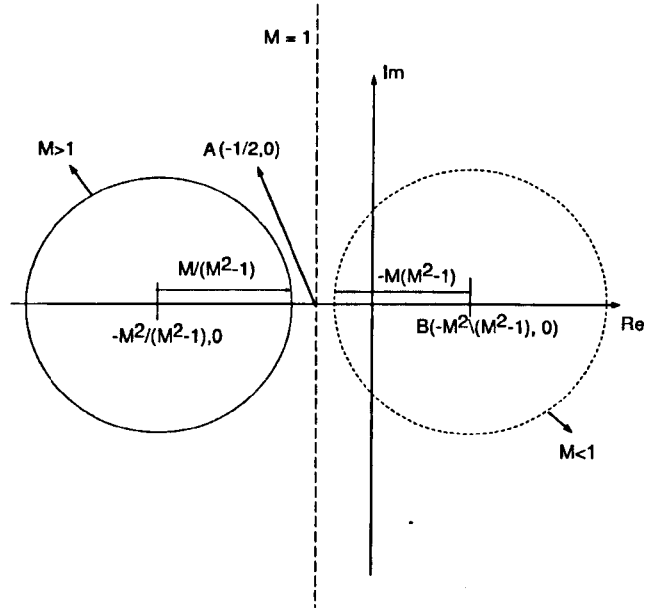


Figure 6.4:  $M$  circle for Case1, Case2 and Case3

We first consider the case  $M > 1$ . In this case the  $M$  contour in the Nichols chart is not defined outside the phase interval  $-270^\circ < \phi < -90^\circ$ . Let  $v = cu$  represent a constant phase line. Solving simultaneously with the equation of the  $M$  circle gives the quadratic:

$$(1 + c^2)u^2 + 2\frac{M^2}{M^2 - 1}u + \frac{M^2}{M^2 - 1} = 0 \quad (6.1)$$

It follows that there are two real roots (actually negative) if

$$-\frac{1}{\sqrt{M^2 - 1}} < c < \frac{1}{\sqrt{M^2 - 1}}$$

corresponding to the case where the line crosses the  $M$ -circle (twice). The conditions

$$c = -\frac{1}{\sqrt{M^2 - 1}} \quad \text{or} \quad c = \frac{1}{\sqrt{M^2 - 1}}$$

correspond to the case when the line  $v = cu$  is tangent to the  $M$  circle, while for  $c > 1/\sqrt{M^2 - 1}$  the solutions of the quadratic are complex conjugate and thus there is no intersection between the line and the  $M$ -circle. Let  $\phi$  denote the phase variable of the polar plot and write  $\phi = -\pi + \theta$  where  $\theta$  denotes the angle with respect to the negative real axis. Let  $\theta_{max}$  denote the angle of the tangent to the  $M$  circle in the second quadrant. Then using the expressions for the co-ordinates of the centre and the

radius of the  $M$  circle we get:

$$\sin \theta_{max} = \frac{\frac{M}{M^2-1}}{\frac{M^2}{M^2-1}} = \frac{1}{M}$$

Note that:

$$\cos \theta_{max} = \frac{\sqrt{M^2-1}}{M} \quad \text{and} \quad \tan \theta_{max} = c = \frac{1}{\sqrt{M^2-1}}$$

the later corresponding to the extreme value of the gradient  $c$  at the tangent condition. Thus in the Nichol's chart the  $M$ -contour is defined for the phase range  $\phi = -\pi + \theta$ ,  $|\theta| \leq \theta_{max} = \sin^{-1}(1/M)$ .

Next consider equation 6.1. In the case that two real solutions exist, the distance of the (two) points of intersection of the circle and the line  $v = cu$  can be calculated as  $d^2 = u^2 + v^2 = (1 + c^2)u^2$  or

$$d_1 = \sqrt{1 + c^2}|u_1| \quad \text{and} \quad d_2 = \sqrt{1 + c^2}|u_2|$$

where  $u_1$  and  $u_2$  represent the two (real-negative) roots. Thus,

$$d^2 = \frac{2M^2}{M^2-1}|u| - \frac{M^2}{M^2-1} = \frac{M^2}{M^2-1}(2|u| - 1)$$

Substituting from the equation  $|u| = d \cos \theta$ , gives:

$$d^2 - \frac{2M^2}{M^2-1} \cos \theta d + \frac{M^2}{M^2-1} = 0$$

which can be solved as:

$$d_{1,2} = \frac{M^2}{M^2-1} \cos \theta \pm \sqrt{\frac{M^4}{(M^2-1)^2} \cos^2 \theta - \frac{M^2}{M^2-1}}$$

or alternatively

$$d_{1,2} = \frac{M^2}{M^2-1} \left( \cos \theta \pm \sqrt{\frac{1}{M^2} - \sin^2 \theta} \right)$$

in the range  $|\theta| < \theta_{max}$ . Thus, in the phase range  $\phi = -\pi + \theta$ ,  $|\theta| \leq \sin^{-1}(1/M)$ , the  $M$ -contour in the Nichol's chart consists of a closed-contour which can be represented as the union of the graphs of the two functions:

$$M_+ = 20 \log_{10} \left( \frac{M^2}{M^2-1} \right) + 20 \log_{10} \left( \cos \theta + \sqrt{\frac{1}{M^2} - \sin^2 \theta} \right)$$

and

$$M_- = 20 \log_{10} \left( \frac{M^2}{M^2-1} \right) + 20 \log_{10} \left( \cos \theta - \sqrt{\frac{1}{M^2} - \sin^2 \theta} \right)$$

Next consider the case  $M = 1$ ; in this case the  $M$ -circle in the Nyquist plane reduces to the straight line  $u = -\frac{1}{2}$ . Thus in the Nichols chart the  $M$ -contour is defined only for the phase interval

$$\phi = -\pi + \theta, \quad |\theta| < \frac{\pi}{2}$$

where

$$d \cos \theta = \frac{1}{2} \quad \text{or} \quad d = \frac{1}{2} \sec \theta$$

Thus the  $M = 1$  contour in the Nichols chart is given as:

$$M = 20 \log_{10} \left( \frac{\sec \theta}{2} \right)$$

This is symmetric with respect to the phase line  $\phi = -\pi$  and tends to infinity as  $|\theta| \rightarrow \frac{\pi}{2}$ .

The final case is when  $M < 1$ . In this case the  $M$ -circle in the Nyquist plane contains the origin. Hence any directed half-line drawn from the origin will intersect the  $M$ -circle just once. We conclude that in this case the  $M$ -contour defined on the Nichols chart is (an open contour) defined for every phase variable. Further it is symmetric with respect to the  $\phi = -\pi$  phase line. Consider again equation 6.1 which can be written as:

$$(1 + c^2)u^2 - \frac{2M^2}{1 - M^2}u - \frac{M^2}{1 - M^2} = 0$$

Note that this time the constant term is negative and therefore we always have two real solutions of opposite sign (one positive and one negative), corresponding to the two intersections of the circle and the line to the right and left of the  $u$ -axis, respectively. The solution of the equation is given as:

$$u_{1,2} = \frac{1}{1 + c^2} \left( \frac{M^2}{1 - M^2} \pm \frac{M^2}{1 - M^2} \sqrt{1 + (1 + c^2) \frac{1 - M^2}{M^2}} \right)$$

or

$$u_{1,2} = \frac{1}{1 + c^2} \frac{M^2}{1 - M^2} \left( 1 \pm \frac{1}{|\cos \phi|} \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

Using the fact that  $1/(1 + c^2) = \cos^2 \phi$  we get

$$u_{1,2} = \frac{M^2 |\cos \phi|}{1 - M^2} \left( |\cos \phi| \pm \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

Next note that:

$$d = |u| \sqrt{1 + c^2} = \frac{|u|}{|\cos \phi|}$$

and hence

$$d_{1,2} = \frac{M^2}{1 - M^2} \left| \cos \phi \pm \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right| = \frac{M^2}{1 - M^2} \left( \sqrt{\frac{1}{M^2} - \sin^2 \phi} \pm |\cos \phi| \right)$$

For  $\phi \in (-2\pi, -\frac{3\pi}{2}) \cup (-\frac{\pi}{2}, 0)$ ,  $|\cos \phi| = \cos \phi$  and hence

$$d_{max} = \frac{M^2}{1 - M^2} \left( \cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

For  $\phi \in (-\frac{3\pi}{2}, -\frac{\pi}{2})$ ,  $|\cos \phi| = -\cos \phi$  and hence

$$d_{min} = \frac{M^2}{1 - M^2} \left( \sqrt{\frac{1}{M^2} - \sin^2 \phi} - |\cos \phi| \right) = \frac{M^2}{1 - M^2} \left( \cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

Thus the  $M$ -contour in the Nichol's chart in this case is

$$M(\phi) = 20 \log_{10}(M^2) - 20 \log_{10}(1 - M^2) + 20 \log_{10} \left( \cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

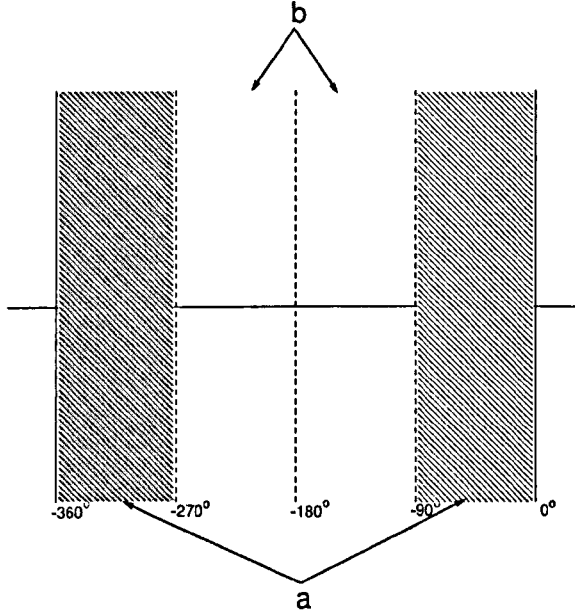


Figure 6.5: Phase regions on the Nichol's chart

### 6.6.2 Algorithm Pseudo-Code

The pseudo-code of the algorithm for constructing  $M$ -circles in the Nichol's chart is as follows:

- Case 1:  $M > 1$ .

- For every phase point check if the phase lies in region *a* as shown in figure 6.5. If yes, then the *M*-contour is not defined in this phase range.
- If the above condition fails then the *M*-contour is defined for a phase interval in this range (symmetric with respect to  $-\pi$  rads). Calculate the gradient *c* of the constant phase line, given by  $c = \tan(\pi + \phi)$ .
- Determine if the *M*-circle intersects the constant phase line, i.e. whether  $|c| \leq \frac{1}{\sqrt{M^2-1}}$ . In this case form the quadratic equation:

$$(1 + c^2)u^2 - 2\frac{M^2}{1 + M^2}u + \frac{M^2}{1 - M^2} = 0$$

and calculate its roots. In general, for  $M > 1$  there may be two (negative), one (double negative), or no real roots depending on the phase.

- Case 2:  $M = 1$ . Here the *M* circle in the Nyquist plane is a vertical line passing through the point  $(-1/2, 0)$ .
- If the phase is in region *a* as shown in figure 6.5 the *M*-contour in the Nichol's chart is undefined at this phase.
- If this condition fails, i.e. the phase is between  $-270^\circ$  and  $-90^\circ$  then set  $M(\phi) = 20 \log_{10}(\sec(\phi + \pi)/2)$ .
- Case 3:  $M < 1$ .
- *M*-contour is open and defined for every phase point. Set

$$M(\phi) = 20 \log_{10}(M^2) - 20 \log_{10}(1 - M^2) + 20 \log_{10} \left( \cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right)$$

The *M*-circles plotted on the Nichol's chart for  $M = 1.2$ ,  $M = 1$ , and  $M = 0.8$  are shown in figure 6.6, the graph was generated using Matlab code provided in Appendix C section 9.3.7.

**High Frequency Bounds:** Upon obtaining the *M*-circles, the next task is to plot the *High frequency bounds*. The algorithm computes the single high frequency boundary for which closed-loop variations are bounded at high frequencies as specified by the specific *M*-circle value specified by the designer.

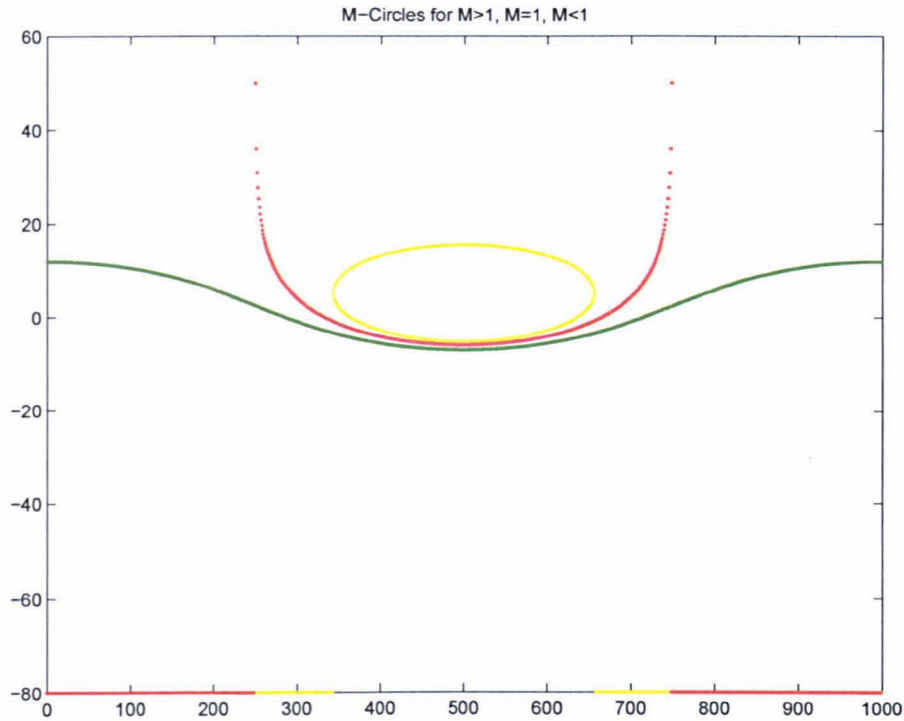


Figure 6.6:  $M$  circle on Nichol's Plot for  $M = 1.2$ ,  $M = 1$  and  $M = 0.8$

### 6.6.3 Algorithm Pseudo-Code

The algorithm produces the system bound for the high frequency region for the closed-loop plant response. The input data required for the design are:

- $M$ -circle value: Closed-loop magnitude bound.
- Maximum level of uncertainty template at infinity.
- Minimum level of uncertainty template at infinity.
- Nominal level of uncertainty template at infinity.

The output from the algorithm will be complex vector containing the calculated high-frequency contour. Upon the availability of the input data the algorithm performs the following tasks:

- Calculate the upper bound limit of uncertainty template at infinity.
- Calculate the lower bound limit of uncertainty template at infinity.
- Obtain the required  $M$ -circle in the Nichol's chart ( $M > 1$  typically).
- Get the maximum and minimum phase limits corresponding to the required  $M$ -

circle.

- Split the  $M$ -circle into upper and lower parts.
- Extend the lower part of the  $M$ -circle by the value corresponding to the upper spread of uncertainty template at infinity, i.e.

$$\begin{aligned}\Delta \log P &= \lim_{\omega \rightarrow \infty} [\log |(P)_{max}(j\omega)| - \log |(P)_{min}(j\omega)|] \\ &= VdB\end{aligned}$$

- Translate the upper part of the  $M$ -circle by the value of the lower spread of the uncertainty template at infinity ( $V$ ).
- Plot the final closed contour.

Matlab code provided in Appendix C section 9.3.8

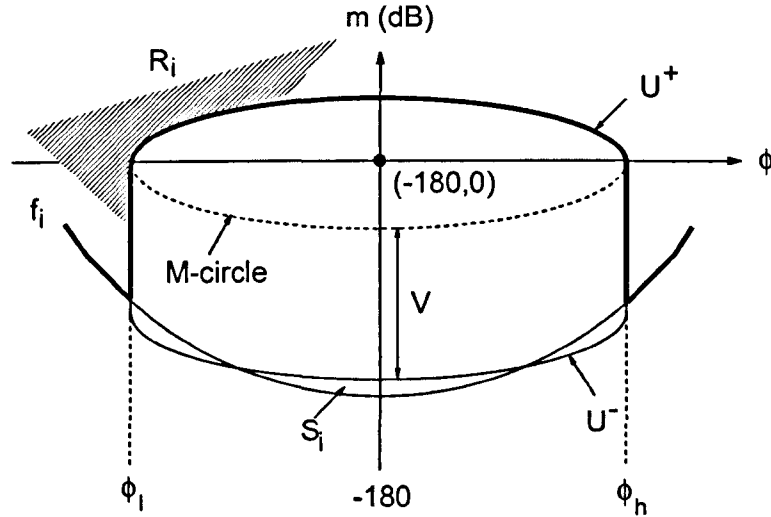


Figure 6.7:  $M$ -circle and  $\mathcal{U}$ -contour in Nichol's chart ( $M > 1$ )

## 6.7 Graphical Stability

The most commonly used method to determine the stability of a closed-loop system using the open-loop response of the system is the *Nyquist stability criterion*. Using this principle this method can be transformed into a stability criterion on the *Nichol's chart*, since *Nichol's chart* is the standard domain used in *QFT* design.

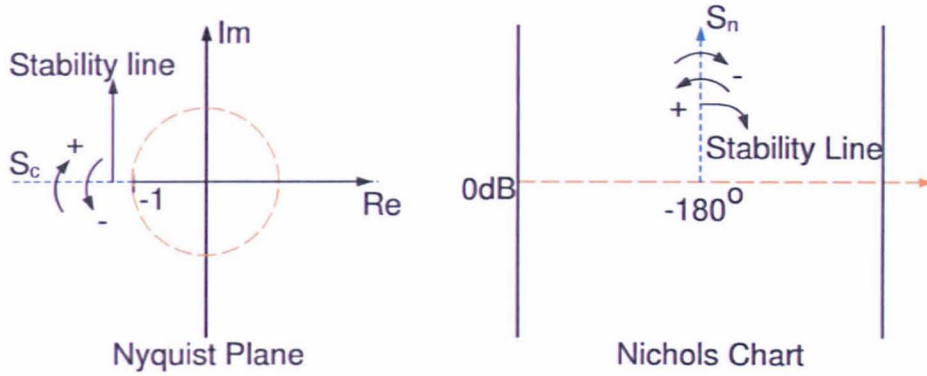


Figure 6.8: *Positive and Negative crossing, and stability lines  $S_c$  and  $S_n$  on the Nyquist plane and Nichol's chart*

## 6.8 Stability criterion

### 6.8.1 Nyquist Stability criterion

The Nyquist stability criterion relates the total number of encirclements of the open-loop frequency response around the critical  $(-1)$  point to the number of the system's closed-loop poles that lie in the right half of the  $s$  - plane. If  $G(s)$  is stable, then its unity feedback closed-loop system is also stable if and only if, the Nyquist contour in the  $G(s)$  plane does not cross or encircle the  $(-1, 0)$  point. If  $G(s)$  has  $P$  poles in the right-half of the  $s$ -plane, then the number of counter-clockwise encirclements of the  $(-1, 0)$  point must be equal to  $P$  for a closed-loop system to be stable [58].

### 6.8.2 Stability criterion on the Nichol's Chart

Since QFT design is carried out on the Nichol's plane, it is sensible to have a criterion to specify stability of a system directly in this plane. A version of Nyquist stability criterion on the Nichol's chart was first developed in [43], and this was further improved in [12].

The stability criterion on the Nichol's chart is a re-formulation of the Nyquist stability criterion. Stability of a system in the Nyquist plane is mainly based in the *Zero exclusion theorem*, i.e. for a system to be stable the condition  $1 + L(j\omega) \neq 0$  should hold for all  $\omega$ , and the net encirclements of the critical point  $-1$  should be zero, i.e.



$N = P - Z$  where  $P$  is the number of RHP poles and  $Z$  is the number of RHP zeros of the system. In the Nyquist plot the direction of the system response produced by an unstable pole is in the *Anti Clockwise* direction in the left half of the Nyquist - plane and this is called *Negative crossing* of the stability line  $S_c$ . The response produced by a zero in the system is in the *Clockwise* direction in the left half of the Nyquist - plane, and this is called *Positive crossing* of the stability line  $S_c$ . The stability line together with the positive and negative crossings are shown in figure 6.9.

First the stability line  $S_c$  which contains the critical point  $-1$  in the Nyquist plane is translated onto the Nichol's chart. The stability line in the Nyquist plane is given by

$$S_c =: \{(x, y) : y = 0, x < -1\} \quad (6.2)$$

$S_c$  is then translated on to the Nichol's chart using the relations  $Z = a + jb$ , where  $a = r \cos \theta$  and  $b = r \sin \theta$ , (in this case  $a = -1$  and  $b = 0$ ), and so the stability line on the Nichol's chart  $S_n$  is given by

$$S_n =: \{(\theta, r) : \theta = -180, r > 0\} \quad (6.3)$$

The stability lines on the Nyquist plane and on the Nichol's chart are shown in figure 6.9. The stability criterion on the Nichol's chart is formulated by interpreting the Nyquist stability criterion in Nichol's chart's terms. The stability line in the Nichol's chart is given by  $S_n$ , this line corresponds to the stability line  $S_c$  in the Nyquist chart as shown in figure 6.9. The stability analysis of a plant in the Nichol's chart is based on the number of positive and negative crossing's of the stability line  $S_n$  by the open-loop response of the system.

To introduce the stability condition a stable system with  $n$  stable poles is considered. Let:

$$L(s) = \frac{N(s)}{D(s)} \quad (6.4)$$

The poles of the above system are lie in the open left half of the  $s$  - plane. The closed-loop stability of the system for various cases is guaranteed by the conditions below:

- For a system whose response lies above the line  $r = 0$  dB the open-loop response

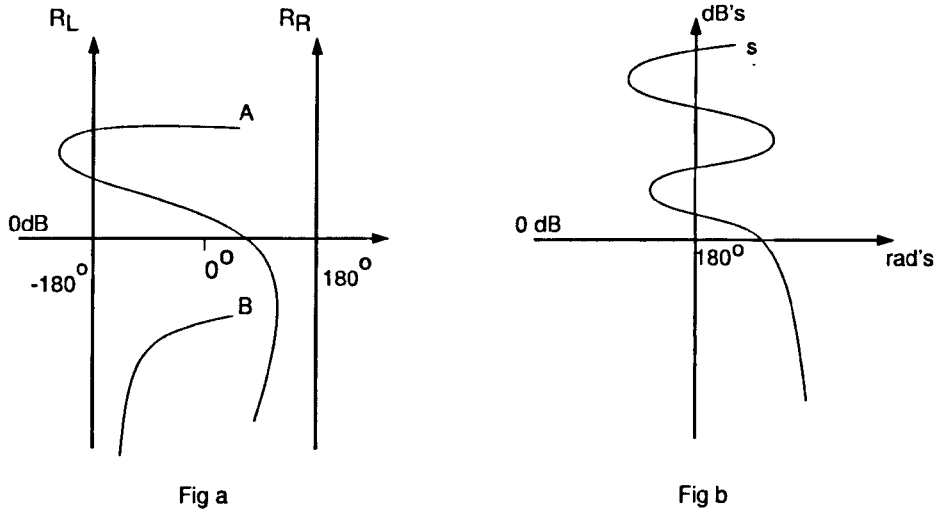


Figure 6.9: *Stability region given on a Nichol's chart*

should pass through the line  $r = 0$  dB, in the range  $-180^\circ$  to  $180^\circ$  to make the system response stable.

- As a direct consequence of the Nyquist stability criterion, the net positive and negative crossings of the stability line  $R_L =: \{\phi = -180^\circ, r = [0, \infty)\}$  and  $R_R =: \{\phi = 180^\circ, r = [0, \infty)\}$ , should be zero, i.e. if the system response crosses the stability lines  $R_L$  or  $R_R$ , then, in order for the system to be stable, the response should re-enter the stability region. Then the response should cross the line  $r = 0$  dB within the stability lines, thus satisfying the above condition.
- For a plant whose open-loop response starts below the line  $r = 0$  dB, the system is always stable.

## 6.9 Graphical design

### 6.9.1 Nyquist Criterion

Although we are only interested in graphical design based on the *Nichol's criterion*, it is useful to know about the graphical design methods related to the *Nyquist criterion*. A graphical Nyquist stability test was proposed in [43]. This method essentially tries to eliminate or pair the positive and negative crossings shown in fig 6.9 on the stability

line in the Nyquist plot, as shown by the authors of [43] using “homotopy equivalence arguments”. The same argument is used here translated to the *Nichol’s chart*.

### 6.9.2 Nichol’s Criterion

The Nichol’s stability criterion proposed in [13] and [43] modifies the well known Nyquist stability criterion onto Nichol’s chart. Although a graphical stability test is relatively straightforward to apply when the controller is designed manually in a graphical environment, automating the test for optimisation based QFT designs poses a number of practical problems. These are associated with the discrete nature of the frequency response necessarily used for loop-shaping, phase discontinuities when the response is limited to the  $(-360^\circ, 0]$  phase interval, etc. In general terms, the graphical design technique proposed here is similar to the graphical Nyquist technique introduced above.

### 6.9.3 Algorithm Pseudo-Code

The input data required to carry out the design are

- num, den: System transfer function numerator and denominator polynomials corresponding to the (nominal) open loop system.
- freq: User specified frequency-array (row-vector) in rads/s.
- tol: Denominator coefficients are perturbed by tol so that generically there are no  $j\omega$ -axis poles.

Based on the conditions described above the algorithm returns a logical flag (either a 0 or a 1) as output. Upon the availability of the required data the following algorithm is implemented.

- Find the dimensions of the denominator polynomial and of the array of user specified frequencies.
- perturb the denominator’s coefficients so that no poles lie on the imaginary axis.
- Normalize the data to ensure that all phase variables are between  $-360^\circ$  and  $0^\circ$ .

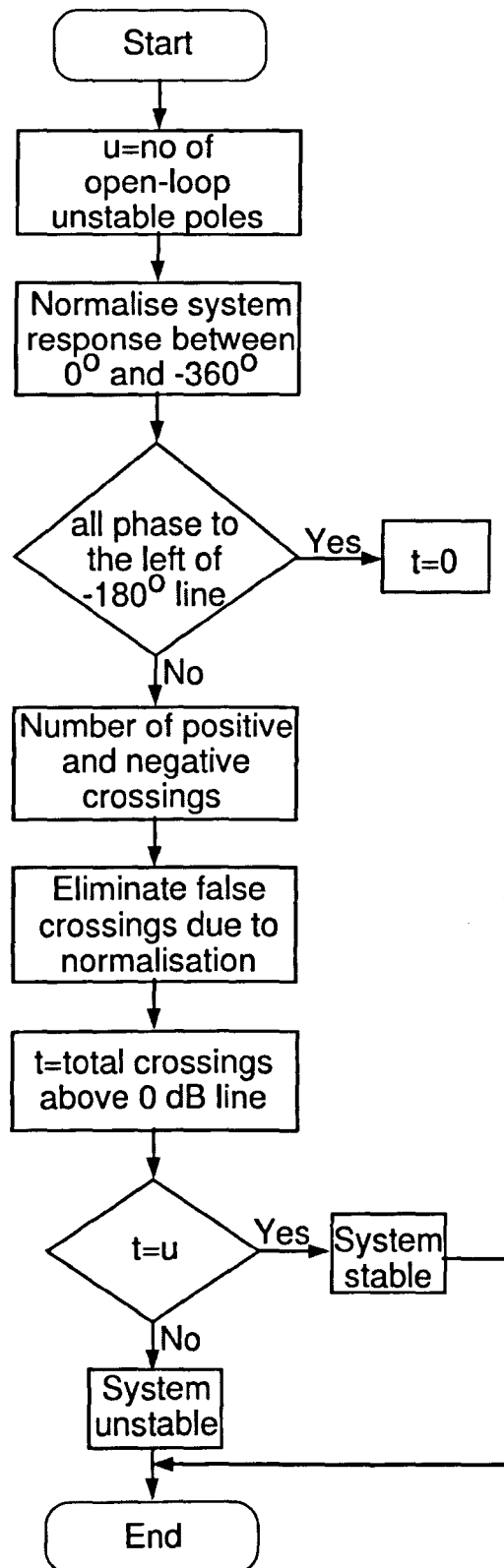


Figure 6.10: *Graphical test flowchart*

- If the minimum magnitude value of the data from the normalised Nichol's plot is less than the tolerance specified by the user, then the system is considered to be unstable.
- Obtain the magnitude, phase and the corresponding frequency values of the plot above the 0 dB line.
- Find the points where any transitions (crossings) occur and their corresponding nature.
- Find any phase discontinuity.
- Eliminate false crossings due to discrete data representation of the frequency response, tangency or phase discontinuity conditions.
- If the number of crossings is equal number of unstable poles then the system is stable, else unstable.

Matlab code provided in Appendix C section 9.3.9

## 6.10 Loop Shaping

The design of the feedback controller is the most important step of the QFT design procedure. It is traditionally performed manually, i.e. via a trial and error procedure. The objective is to shape the nominal open-loop frequency response of the system so that:

- It lies above the robust-performance contours,
- It avoids the low-damping region ( $U$ -contour), and
- It encircles the critical point in the anti-clockwise direction  $p$  times, where  $p$  is the number of open-loop unstable poles.

It is implicitly assumed that the whole family of uncertain plants has the same number of unstable poles, and that no unstable pole/zero cancelations occur between the feedback controller and  $G(p, s)$  for every  $p \in \mathcal{P}$ . Although this last condition requires independent verification, it is rarely an issue in practice designs since only stable and

minimum-phase controllers are typically employed.

Among all possible controllers which meet the above requirements, the “best” design is considered to be the one in which the open-loop frequency response at the design frequencies lies as close as possible to the robust performance templates. This is in order to avoid “over-designing” the system by using excessively large gains, which may lead to noise amplification, instability due to un-modelled dynamics, etc. Very sensibly, the QFT design philosophy demands the use of “as little feedback as possible” consistent with the robust performance specifications.

The software provides an interactive design tool to help with manual loop-shaping and three optimisation algorithms for automatically designing “optimal” controllers which meet the QFT constraints described above. These are:

- An algorithm for designing optimal fixed-structure controllers (PID, first-order lead/lag, second-order with complex poles),
- An algorithm for designing controllers in the frequency-domain via linear programming, and
- An convex optimisation algorithm for designing the optimal controller indirectly by shaping the closed-loop transfer function

The manual loop-shaping tool and the algorithm for designing fixed-structure controllers is described in more detail in the following sections. The last two algorithms have not been included in the toolbox as they rely on many design-specific details which are difficult to include systematically in a generic software tool. However, their main characteristics have been described at various sections in the thesis and software routines that can be used for their implementation are included in the Appendix C.

### **6.10.1 Manual loop-shaping tool**

This is an interactive graphical design tool for shaping the open-loop characteristics of the system according to the QFT constraints described above. By clicking on two points on the Nichol’s chart, the required frequency and phase/gain differences are calculated and an appropriate first-order phase lead/lag controller are automatically calculated. Alternative options include the addition of integral, derivative, proportional or second-

order terms with complex roots. The uncompensated and compensated frequency responses are next displayed on the Nichol's chart. Based on this graphical output the designer can proceed with the design by adding a new factor in the compensator, back-stepping by deleting the last compensator factor designed, etc. The overall logic of the tool summarising the options available to the designer at each stage are shown in the flow-diagram in Figure 6.11.

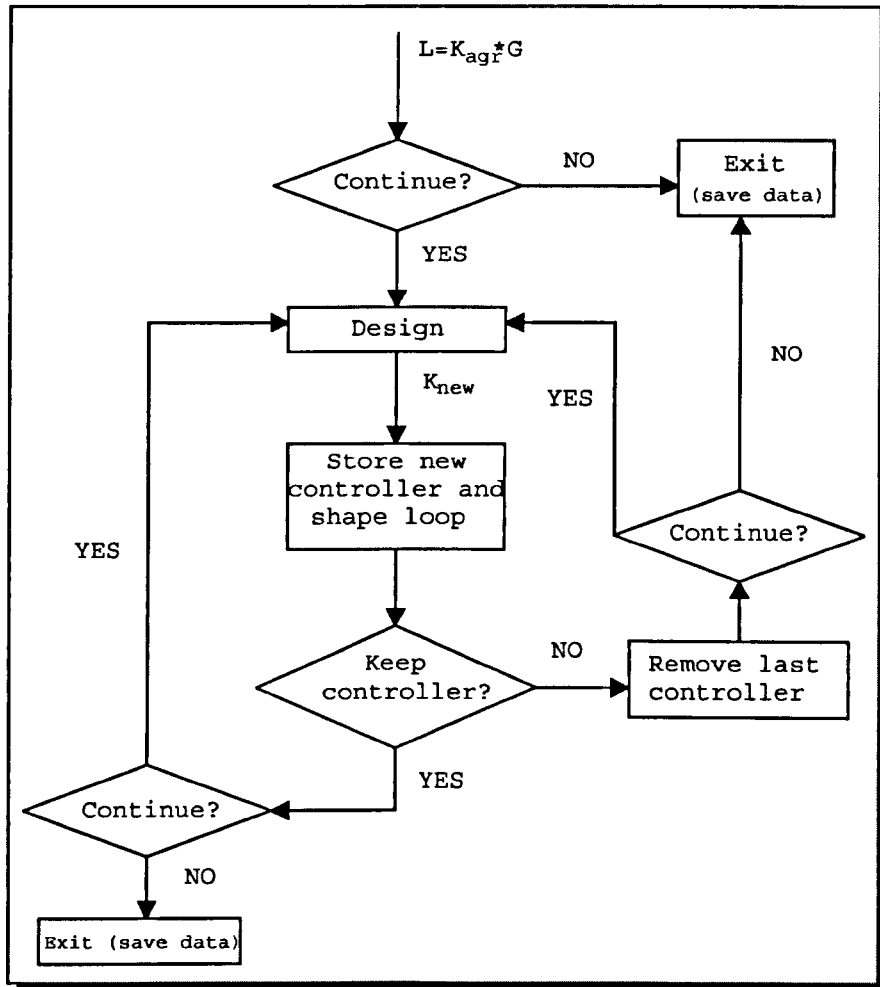


Figure 6.11: Lead/Lag-network manual loop shaping tool

Clearly, there is not a unique way for designing a compensator using this method, and often a successful design requires a considerable amount of experience. When the specifications are tight, the design procedure may require a large number of cascade terms, resulting in a high-order overall controller. To help with this potential increase in controller complexity, a controller model-reduction tool is supplied. The effectiveness of this reduction procedure may be checked by displaying the frequency responses of the

designed (high-order) system and its low-order approximation on the Nichol's chart.

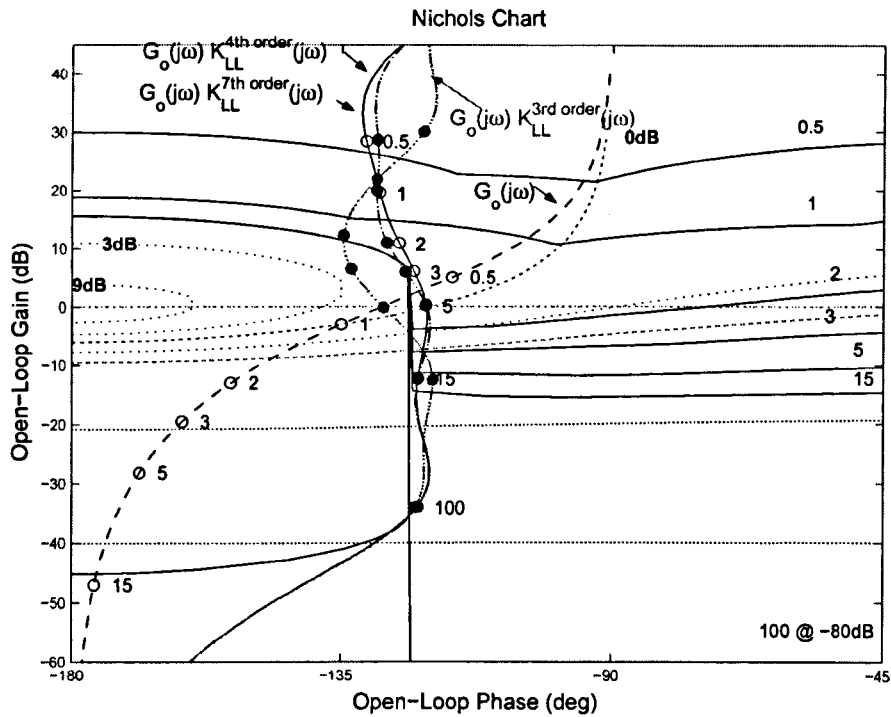


Figure 6.12: Control Design with Lead/Lag Network

The tool was used to design a feedback controller for the system introduced in an earlier chapter. The resulting frequency-response of open-loop system (with the designed 7-th order controller), along with the nominal plant are displayed in Figure 6.12. It may be seen that all robust stability and performance specifications are satisfied; in addition the open-loop frequency response points at the seven design frequencies are reasonably close to the corresponding robust performance templates. When controller model reduction was performed it was found that up to 3 states could be removed without affecting significantly the controller's frequency response. The corresponding nominal open-loop response using a 4th-order and a 3rd-order controller are also shown in Figure 4 for comparison.

### 6.10.2 Fixed-structure controller optimisation

The algorithms described in this section are implementations of the methods in [63] and [27]. The optimisation is carried out over the parameters of fixed-structure con-



trollers (PID, first-order lead/lag, second-order). The main idea behind the algorithm is that, for the selected control structures, fixing the phase of the controller at *two* distinct frequencies, determines completely the phase response of the controller over *all* frequencies and thus also the phase of the nominal open-loop system. Thus, using the robust stability and performance bounds on the Nichol's chart, it is straightforward to determine the minimum controller gain (if it exists) so that all robust stability and performance objectives are satisfied. (Note that if the phase of the nominal open-loop system is fixed, varying its gain corresponds to shifting the response vertically in the Nichol's chart). Repeating the procedure over all phase combinations (suitably discretised) will produce the optimal controller parameters. Optimality in this context may be defined in terms of asymptotic open-loop gain, nominal/worst-case cross-over frequency or closed-loop bandwidth, or some other appropriate measure consistent with the general QFT philosophy penalising system "over-design".

The numerical techniques used to implement the algorithm include: (a) Singular value decomposition (for phase-lead/lag compensation only), and (b) A robust stability gain margin calculation. This can be performed via either a numerical algorithm (when the nominal transfer function is known) or via purely graphical means (i.e. directly from the frequency response) by counting the crossings on a certain line on the Nichol's chart and its directions (see [13], [12] for details).

The nominal open-loop frequency response corresponding to the optimal PID controller for a previous example is shown in Figure 6.13 below. Note that the gain for all design frequencies lies above the corresponding robust performance bounds, with one lying exactly on the robust stability boundary ( $\mathcal{U}$ -contour). The asymptotic phase of the loop is  $-90^\circ$  since the controller includes a pure derivative term. A simple modification to the algorithm is required if one wishes to limit the derivative action at high frequencies. The algorithms described in this section may be used to design simple controllers or as a first step in a more complex design. Note that every rational controller of arbitrary complexity can be constructed from cascade interconnections of the types used here (i.e. integrator, phase lead/lag, second-order denominator or numerator term). Thus, it is possible to improve the design continuously by building high-order controllers in a step-by-step procedure: At each step the optimisation algorithm is carried out (for

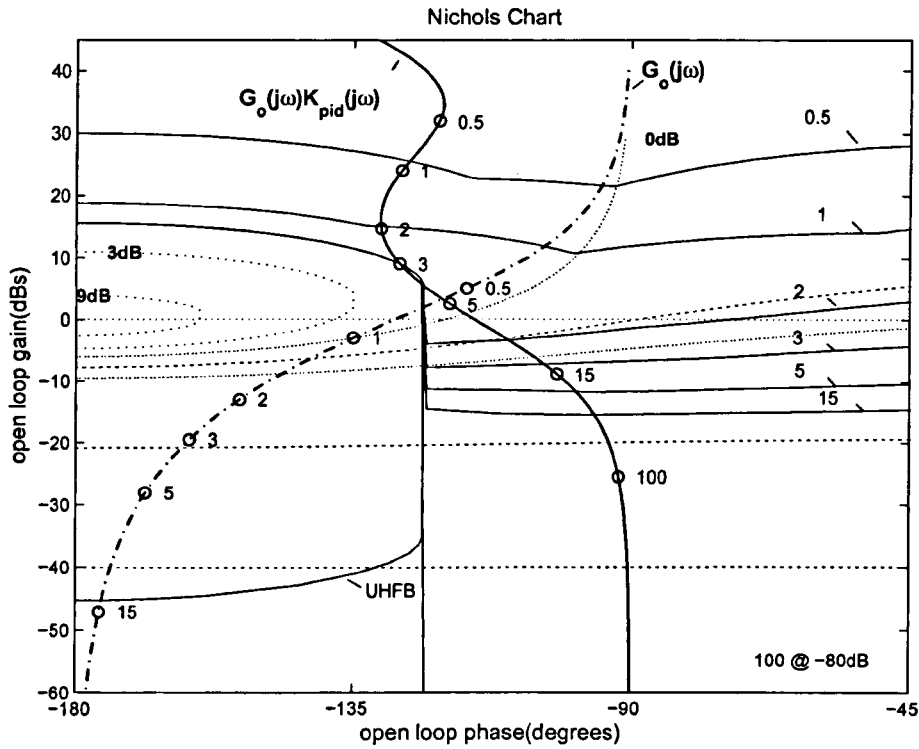


Figure 6.13: Design with Optimal PID controller

one of the above three structures) and the resulting optimal controller  $K(s)$  is accumulated into the nominal open-loop system by redefining  $L_o(s) \leftarrow L_o(s)K(s)$ . This process may continue until a satisfactory design is obtained, or until the “cost” fails to decrease significantly. Of course, the controller resulting from this procedure will not, in general, be optimal over the higher-order controller set. Matlab code to design the optimal controller is provided Appendix C section 9.3.10.

Next, we include the code for the graphical design tool. This is based on the following functions:

- *graphdes.m*: Performs graphical design using arbitrary interconnection of phase lead/lag compensators.
- *create.m*: Subroutine of *graphdes.m* - Creates data for QFT design.
- *gr-des.m*: Subroutine of *graphdes.m* - Low-lever graphical design tool of phase lead/lag compensator.

- *ph-lead.m*: Subroutine of gr-des.m - Phase lead design routine.
- *ph-lag.m*: Subroutine of gr-des.m - Phase lag design routine.

Matlab code for the graphical design tool provided in Appendix C section 9.3.11

## 6.11 Pre-Filter

The final part of the algorithm is to design the pre-filter  $F(s)$  shown in Fig 1.1. The purpose of the pre filter is to restrict the responses of the complementary sensitivity function  $\log \left| \frac{PG}{1+PG} \right|$  on the Bode plot to lie within the tracking specifications  $T_{Ru}$  and  $T_{RL}$ . The algorithm is provided information about the plant, the controller and the cutoff frequencies. The output of the algorithm is the transfer function of the pre-filter as shown in Fig 1.1. Matlab code for pre-filter design provided in Appendix C section 9.3.12.

## 6.12 Conclusions

The chapter has described a software tool which implements in Matlab's graphical environment the main steps of the QFT design procedure. The tool can be used to design robust controllers for uncertain systems using the design philosophy of the QFT method and also as a test-bed for new techniques in this area, including loop-shaping, which is the most challenging step of the design procedure.

# Chapter 7

## Case study - Design of hydraulic actuator

### 7.1 Introduction

In this chapter a control design case study of a non-linear hydraulic actuator is undertaken, using the developed methods of this thesis, based on the Quantitative Feedback Theory. The model of the actuator represents a real system which is fully described in [40], [41]. The model is linearized around an operating point and the uncertainty in the nominal plant is quantified. This involves ten uncertain parameters which are assumed to vary independently over their corresponding ranges. Using the robust design specifications of [40], a robust QFT controller is designed using a fixed structure optimization method developed in Chapter 5. The design concludes by the design of the pre-filter using a systematic procedure based on linear programming. The designed feedback controller has low complexity and, as shown via numerous simulations, is successful in meeting the design specifications in terms of robust stability and robust performance specifications for the complete set of model uncertainty. The chapter concludes by comparing the designed controller with the one obtained in [40] and summarizing the results which illustrates the effectiveness of the proposed design method.

The chapter is organized as follows: Section 8.2 provides a background of recent research on force control of hydraulic actuators, describes the main issues involved in their design (especially non-linearities and model uncertainty) and the various control methods that have been proposed in the literature. Section 8.3 presents a detailed modelling procedure for a hydraulic actuator interacting with an uncertain environment. Although the modelling techniques presented in this section are rather

standard, emphasis is placed on highlighting all assumptions and for developing an appropriate parametric LTI model, together with a quantitative uncertainty model which is used extensively in the sequel. Section 4 contains the definition of the design objectives and outlines the steps of the QFT-based procedure for designing a feedback controller and a pre-filter that are of low complexity and meet the design specifications. The design of the feedback controller is based on a fixed structure optimization method developed in a previous chapter, while the design of the pre-filter is based on a linear programming optimization algorithm. Both steps are successful in designing the overall compensation scheme, as is illustrated via numerous simulations and a detailed analysis of the results. The feedback controller is compared with the one designed in [40] using a QFT manual loop-shaping procedure and validated experimentally. The two controllers are found to exhibit similar responses, which is not surprising as the design specifications are tight. It is thus possible to conclude that the fixed-structure optimization algorithms developed in this work can automate the loop-shaping step of the QFT design and are useful in practical design situations. The overall conclusions of the Chapter appear in Section 5.

## 7.2 Background and General Design Objectives

A wide range of engineering problems involve the control of hydraulic actuators interacting with uncertain environments. These include flight control, robot position control, manufacturing systems, etc.

Several strategies have been proposed for controlling hydraulic actuators. Some recent approaches proposed in the literature include:

- A combination of velocity feed-forward, output feedback and a Luenberger observer with state estimate feedback for force control. The simulation and experimental results for a constant set point force show superior performance of the proposed method over conventional (P or PI) force feedback controllers [2].
- In [3] a sliding mode controller is designed for a single-rod hydraulic actuator interacting with a spring as an environment. Using position, velocity, acceleration, force, and pressure feedback, the variable-structure controller designed is reported to be suitable for both static and dynamic force control tasks. The ef-

fect of servo-amplifier gain variation was also examined. Reference [4] employed a sliding-mode controller with a perturbation observer for a single-rod electro-hydraulic system. The effect of cylinder position and velocity on the pressure dynamics was considered as perturbation, estimated via an observer. The experimental results demonstrate improved steady-state and transient performance compared to traditional proportional-integral-differential (PID) designs.

- In [5] adaptive control strategies have been considered for hydraulic force control. A switching control scheme was developed using a Lyapunov-based adaptive control law to reduce the effects of parametric uncertainty. The implementation of the controller, which is based on measurements of position, velocity, acceleration, pressure, and spool displacement, showed good performance for high-frequency force/pressure tracking. In Reference [6] a generalized predictive control algorithm was applied to a hydraulic force control system. The controller was experimentally evaluated for various environmental stiffness and set-points. The method, however, relies heavily on online parameter estimation and consequently is computationally expensive.
- Reference [7] uses an  $\mathcal{H}_\infty$  approach to robustly control the force exerted by a double-acting symmetric hydraulic cylinder with a servo-valve. The importance of uncertainties and nonlinearities on the performance of hydraulic force control systems is highlighted. Limited test results, demonstrate the stability performance trade-off of the system.

Despite the existence of a great number of force control concepts, methods, and algorithms, there is still a gap between theory and industrial practice. The main reasons are due to poor industrial control architecture, which does not allow the implementation of complex algorithms. Thus the design of simple controllers, ideally of a fixed structure, becomes an important design issue.

In this chapter QFT techniques are employed to design an fixed-structure force control scheme for an industrial hydraulic actuator model. The goal is to arrive at a fixed-gain controller that:

1. Is low order and easy to implement.

2. Is robust against uncertainties in both environmental stiffness and actuator functions, and
3. Does not require precise knowledge of the systems parameters.

## 7.3 Modelling of Hydraulic actuator interacting with environment

In this section a linear parametric model of the actuator is derived. This follows references [40], [41] which gives a brief outline of the model derivation. Here this material is expanded to include full details. The main simplification assumptions made and the origin of uncertainty in the derived model are highlighted in the presentation.

A schematic of the hydraulic actuator is shown in Figure 7.1. Uncertainty in the model arises from variation in operating-point dependent parameters, changes in the environment and changes in the hydraulic actuator's functions.

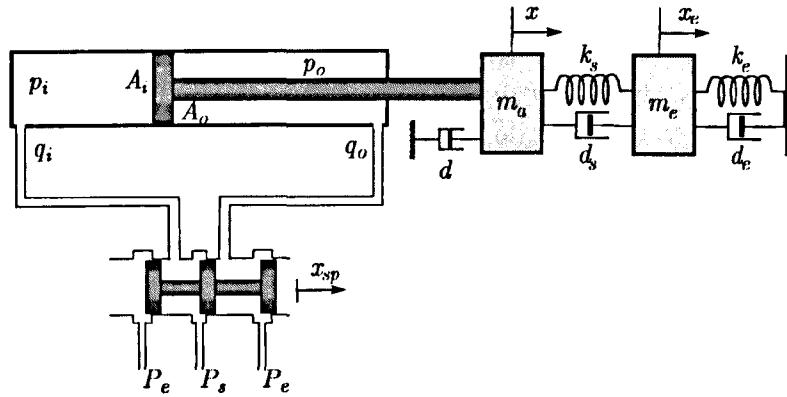


Figure 7.1: Diagram of the hydraulic actuator (based on [40])

We next present a detailed analysis leading to the derivation of the linearised model of the actuator. This is expanded from the work of [40] which does not present a detailed analysis of the modelling procedure.

The schematic of Figure 7.1 can be decomposed into three parts. These are

1. Electric relay
2. Actuator

### 3. Manipulator-sensor-environment.

#### (i) Electric relay

We first consider the model of the electric relay. By supplying power to the electric relay, the lever inside the choke moves. The displacement depends on how much power is supplied (depending on relay specifications) and the direction of movement depends on the polarity of power supply.

Let  $x_{sp}$  denote spool displacement. We consider jointly the two cases of extension ( $x_{sp} \geq 0$ ) and retraction ( $x_{sp} < 0$ ). The relationship of spool displacement and the applied input voltage,  $U$ , can be adequately modelled via the first-order differential equation:

$$U(t) = \frac{\tau}{k_{sp}} \frac{dx_{sp}(t)}{dt} + \frac{1}{k_{sp}} x_{sp}(t)$$

where  $x_{sp}$  represents spool displacement,  $\tau$  is the effective time constant,  $k_{sp}$  is the gain parameter and  $U(t)$  represents input voltage. The transfer function of the system may now be derived by taking Laplace transforms as:

$$\frac{X_{sp}(s)}{U(s)} = \frac{k_{sp}}{1 + \tau s}$$

where  $s$  is the Laplace transform variable.

#### (ii) Actuator

By spool displacement, the effective openings of the valves changes, and thus also the pressure injected to the main piston. We thus have different pressures  $P_i$ ,  $P_o$  on the two sides of the piston and so a differential force  $F_a$  is developed which moves the piston. The governing nonlinear equations describing the fluid flow are written, in their simplest form, in the form shown below. We neglect leakage flow across the actuator's piston and distinguish between the following two cases:

*Case 1:*  $x_{sp} \geq 0$  (extension). Here:

$$q_i = C_b \omega x_{sp} \sqrt{\frac{2}{\rho}} (P_s - P_i)$$
$$q_o = C_b \omega x_{sp} \sqrt{\frac{2}{\rho}} (P_o - P_e)$$



Case 2:  $x_{sp} < 0$  (contraction). In this case:

$$q_i = C_b \omega x_{sp} \sqrt{\frac{2}{\rho}} (P_i - P_e)$$

$$q_o = C_b \omega x_{sp} \sqrt{\frac{2}{\rho}} (P_s - P_o)$$

where  $q_i$  is the fluid flow into the valve,  $q_o$  is the fluid flow out of the valve,  $P_s$  is pump pressure,  $P_e$  is return (exit) pressure,  $\rho$  is the mass density of the fluid and  $\omega$  is the area gradient coefficient that relates spool displacement to orifice area.

The equations above are non-linear and need to be linearized about an operating point to obtain a linear transfer function. As the main objective is to utilize this model within the QFT design framework, the variation of the operating-point dependent parameters will be included in the model as uncertainty. Here, the derivation of the linearised model is described in a step-by-step procedure to highlight the assumptions made during the derivation and simplification of the model.

Flow variables  $q_i$  is a non-linear function of  $x_{sp}$  and  $p_i$ . Similarly, flow variables  $q_o$  depends on  $x_{sp}$  and  $p_o$ . Note that we now make use of the standard notation  $P_o(t) = p_o + \delta p_o(t)$  (and similarly for the other variables), where  $p_o$  is assumed to be a fixed pressure level around which the linearisation is carried out and  $\delta p_o$  represents a small variation around  $p_o$ . Using the chain rule for partial derivatives we get:

$$\delta q_i = \left. \frac{\partial q_i}{\partial x_{sp}} \right|_{(x_{sp}, p_i)} \delta x_{sp} + \left. \frac{\partial q_i}{\partial p_i} \right|_{(x_{sp}, p_i)} \delta p_i$$

$$\delta q_o = \left. \frac{\partial q_o}{\partial x_{sp}} \right|_{(x_{sp}, p_o)} \delta x_{sp} + \left. \frac{\partial q_o}{\partial p_o} \right|_{(x_{sp}, p_o)} \delta p_o$$

The four partial derivatives can now be calculated (at the indicated linearization points) as:

Case (i),  $x_{sp} \geq 0$  (extension):

$$k_s^i := \frac{\partial q_i}{\partial x_{sp}} = C_d \omega \sqrt{\frac{2}{\rho}} (P_s - P_i)$$

$$k_s^o := \frac{\partial q_o}{\partial x_{sp}} = C_d \omega \sqrt{\frac{2}{\rho}} (P_o - P_e)$$

$$k_p^i := \frac{\partial q_i}{\partial p_i} = -\frac{C_d \omega x_{sp}}{\sqrt{2\rho(P_s - P_i)}}$$

$$k_p^o := \frac{\partial q_o}{\partial p_o} = \frac{C_d \omega x_{sp}}{\sqrt{2\rho(P_o - P_e)}}$$

and

Case (ii),  $x_{sp} < 0$  (retraction):

$$k_s^i := \frac{\partial q_i}{\partial x_{sp}} = C_d \omega \sqrt{\frac{2}{\rho}} (P_i - P_e)$$

$$k_s^o := \frac{\partial q_o}{\partial x_{sp}} = C_d \omega \sqrt{\frac{2}{\rho}} (P_s - P_o)$$

$$k_p^i := \frac{\partial q_i}{\partial p_i} = \frac{C_d \omega x_{sp}}{\sqrt{2\rho(P_i - P_e)}}$$

$$k_p^o := \frac{\partial q_o}{\partial p_o} = -\frac{C_d \omega x_{sp}}{\sqrt{2\rho(P_s - P_o)}}$$

Thus, redefining variables ( $p_i \leftarrow \delta p_i$ ), etc, to avoid having increments as state variables) we can write:

$$q_i = k_s^i x_{sp} + k_p^i p_i$$

$$q_o = k_s^o x_{sp} + k_p^o p_o$$

where the coefficients  $k_s^i$ ,  $k_p^i$ ,  $k_s^o$  and  $k_p^o$  represent flow and pressure sensitivity gains of the valve. Here superscripts  $i$  and  $o$  stand for input and output, respectively. Note that the four coefficients are defined differently in the two cases  $x_{sp} \geq 0$  and  $x_{sp} < 0$ . For

modelling simplicity it is now assumed that the system has same response in extension and retraction, i.e. that the numerical values of each pair of coefficients are identical in the two cases, and thus the only distinguishing feature between the two cases is the sign of the coefficient. Then using capital letter notation, we can write the above two expressions in the form:

*Case (i),  $x_{sp} \geq 0$  (extension):*

$$q_i = K_s^i x_{sp} - K_p^i p_i \quad (7.1)$$

$$q_o = K_s^o x_{sp} + K_p^o p_o \quad (7.2)$$

*Case (ii),  $x_{sp} < 0$  (retraction):*

$$q_i = K_s^i x_{sp} + K_p^i p_i \quad (7.3)$$

$$q_o = K_s^o x_{sp} - K_p^o p_o \quad (7.4)$$

Note that now all sensitivity coefficients  $K_s^i$ ,  $K_p^i$ ,  $K_s^o$  and  $K_p^o$  are *positive*.

Next, consider the force developed on the piston due to the presence of different pressures on its two sides. This is given as:

$$F_a = A_i P_i - A_o P_o$$

where  $A_i$  and  $A_o$  are the effective areas (inner and outer), and  $P_i$ ,  $P_o$  represent the inner and outer line pressures, respectively. The inward/outward flows may be expressed as:

$$q_i = A_i \frac{dx}{dt} + \frac{V_i}{\beta} \frac{dP_i}{dt}$$

and

$$q_o = A_o \frac{dx}{dt} - \frac{V_o}{\beta} \frac{dP_o}{dt}$$

where  $x$  denotes the displacement of the piston,  $\beta$  is the effective bulk modulus of the hydraulic fluid and  $V_i$ ,  $V_o$  represent the volumes of the fluid at the two sides of the piston, which are functions of the piston displacement, i.e.  $V_i = V_i(x)$  and  $V_o = V_o(x)$ . Now, for small displacements  $x$ ,

$$V_i = \bar{V}_i + x A_i, \quad V_o = \bar{V}_o + x A_o$$

where  $\bar{V}_i$  and  $\bar{V}_o$  represent the initial volumes of the fluid trapped on the two sides of the piston. Also, assuming small piston displacements in the vicinity of the mid-stroke,

the following approximations can be made:

$$\frac{V_i(x)}{\beta} \approx \frac{V_o(x)}{\beta} \approx \frac{\bar{V}_i + \bar{V}_o}{2\beta} := C$$

and thus,

$$q_i = A_i \frac{dx}{dt} + C \frac{dP_i}{dt}, \quad q_o = A_o \frac{dx}{dt} - C \frac{dP_o}{dt}$$

Using equations (7.1), these can be written in the Laplace transform domain as:

$$P_i(s) = \frac{A_i s}{Cs + K_p^i} X(s) + \frac{K_s^i}{Cs + K_p^i} X_{sp}(s), \quad P_o(s) = \frac{A_o s}{Cs + K_p^o} X(s) + \frac{K_s^o}{Cs + K_p^o} X_{sp}(s)$$

Any variation in parameters  $K_s^i$  ( $K_s^o$ ) and  $K_p^i$  ( $K_p^o$ ) can be included as uncertainty in the new-defined parameters  $K_s$  and  $K_p$ , respectively. Thus, the force acting on the piston can be written as:

$$F_a(s) = \left( \frac{K_s A_o}{Cs + K_p} + \frac{K_s A_i}{Cs + K_p} \right) X_{sp}(s) - \left( \frac{A_i^2 s}{Cs + K_p} + \frac{A_o^2 s}{Cs + K_p} \right) X(s)$$

### (iii) Manipulator-sensor-environment

In this part, the model of the manipulator-sensor-environment is derived, which is coupled to the hydraulic actuator dynamics. A schematic diagram of this part of the overall model is shown in the figure below:

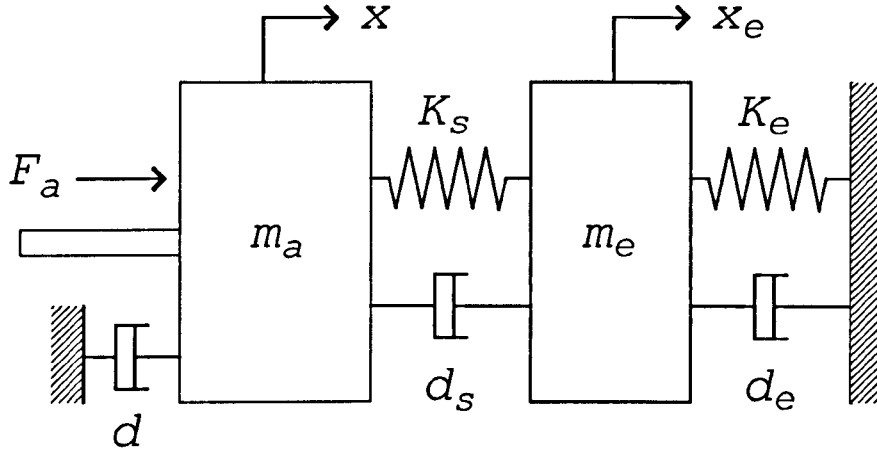


Figure 7.2: Manipulator-sensor-environment

Here  $K_s$  and  $K_e$  represent sensor and environment stiffness,  $d_s$  and  $d_e$  represent sensor and environment damping and  $d$  models relative damping between  $m_s$  (sensor

mass) and  $m_e$  (environment mass). The sensed force is measured via the elongation (contraction) of spring  $K_s$ , i.e.

$$F = k_s(x - x_e)$$

The force  $F_a$  is applied to the mass  $m_a$ . The sliding friction between the masses  $m_a$  and  $m_e$  surface is assumed negligible. The system equation can be written in terms of the two displacements  $x$  and  $x_e$ . The mechanical network is drawn by connecting the terminals of the elements that have the same displacement. So, we will have two sub-networks, and the equations of motion can be written for each one of them.

Force balancing for the first sub-system gives:

$$F_a = m_a \ddot{x} + d_s(\dot{x} - \dot{x}_e) + d\dot{x} + k_s(x - x_e)$$

Taking Laplace transforms gives:

$$F_a(s) = [m_a s^2 + (d + d_s)s + k_s]X(s) - (d_s s + k_s)X_e(s)$$

Force balancing for the second subsystem gives:

$$m_e \ddot{x}_e - d_s(\dot{x} - \dot{x}_e) + d_e \dot{x}_e - k_s(x - x_e) + k_e x_e = 0$$

and hence,

$$X(s) = \frac{m_e s^2 + (d_s + d_e)s + k_s + k_e}{d_s s + k_s} X_e(s)$$

#### (iv) Overall model

Combining the various equations obtained so far allows us to determine the overall model of the system. First, dividing equations for  $F(s)$  and  $F_a(s)$  gives:

$$\frac{F_a(s)}{F(s)} = \frac{[m_a s^2 + (d_a + d)s + k_s]X(s) - (d_s s + k_s)X_e(s)}{k_s(X(s) - X_e(s))}$$

which may be re-written as:

$$\frac{F_a(s)}{F(s)} = \frac{\left( \frac{m_a s^2 + (d_a + d)s + k_s}{d_s s + k_s} \right) (m_a s^2 + (d_s + d)s + k_s)X_e(s) - (d_s s + k_s)X_e(s)}{k_s \left( \frac{(m_e s^2 + (d_s + d_e)s + (k_s + k_e))}{d_s s + k_s} - 1 \right) X_e(s)}$$

which may be simplified to

$$\frac{F_a(s)}{F(s)} = \frac{1}{k_s} \left( m_a s^2 + (d + d_s)s + k_s + \frac{(d_s s + k_s)(m_a s^2 + d_s s)}{m_e s^2 + d_e s + k_e} \right)$$

after some algebra. Using the second equation of the manipulator-sensor-environment dynamics gives:

$$\frac{X(s)}{F(s)} = \frac{1}{k_s} \frac{m_e s^2 + (d_s + d_e)s + k_s + k_e}{m_e s^2 + d_e s + k_e}$$

To summarise, the following set of equations will be used to derive the transfer function of the system:

$$F_a = \frac{K_s A_o + K_s A_i}{C s + K_p} X_{sp} - \frac{(A_i^2 + A_o^2)s}{C s + K_p} X(s)$$

$$\frac{F_a(s)}{F(s)} = \frac{1}{k_s} \left( m_a s^2 + (d + d_s)s + k_s + \frac{(d_s s + k_s)(m_a s^2 + ds)}{m_e s^2 + d_e s + k_e} \right)$$

$$X(s) = \frac{m_e s^2 + (d_s + d_e)s + k_s + k_e}{d_s s + k_s} X_e(s)$$

Using these three equations, the transfer function between  $X_{sp}$  and  $F$  can be obtained as:

$$\frac{F(s)}{X_{sp}(s)} = \frac{K_s k_s (A_i + A_o)}{(C s + K_p) (D_1) + s(A_i^2 + A_o^2) \left( 1 + \frac{d_s s + k_s}{m_e s^2 + d_e s + k_e} \right)}$$

Where

$$D_1 = m_a s^2 + (d + d_s)s + k_s + \frac{(d_s s + k_s)(m_a s^2 + ds)}{m_e s^2 + d_e s + k_e}$$

The equation can be further simplified if the stiffness of the force sensor and the piston rod are high compared with the environmental stiffness and the hydraulic compliance. Hence their dynamics are excited during the contact and can be lumped together as a right body. Based on this assumption we set  $d_s = d_e = 0$  and  $m_e = 0$  and the above equation simplifies as:

$$\frac{F(s)}{X_{sp}(s)} = \frac{k_e K_s (A_i + A_o)}{(C s + K_p) \left( \frac{m_a k_e}{k_s} s^2 + \frac{d k_e}{k_s} s + k_e + \frac{s k_s (m_a s d)}{k_e} \right) + s \frac{k_e}{k_s} (A_i^2 + A_o^2) \left( 1 + \frac{k_s}{k_e} \right)}$$

Under the additional assumption that  $k_s \gg k_e$  the system transfer function can be simplified further, as shown below:

$$\frac{F(s)}{X_{sp}(s)} = \frac{k_e K_s (A_i + A_o)}{(C s + K_p) (m_a s^2 + ds + k_e) + s(A_i^2 + A_o^2)}$$

Introducing as the input variable the voltage  $V(s)$  rather than  $x_{sp}$ , produces the (final) transfer function of the hydraulic actuator between applied voltage  $V(s)$  and measured

contact force  $F(s)$  in the form:

$$G(s) = \frac{F(s)}{V(s)} = \frac{k_{sp}}{\tau s + 1} \left[ \frac{K_s k_e (A_i + A_o)}{(K_p + Cs)(m_a s^2 + ds + k_e) + (A_i^2 + A_o^2)s} \right] \quad (7.5)$$

which will be used in all subsequent analysis.

## 7.4 Uncertainty modelling and QFT Control Design

The transfer function of the hydraulic actuator is defined in equation (7.5). The variables used in this formula are summarize next for convenience: Variables  $k_s$  and  $d_s$  represent sensor stiffness and damping, respectively. The sensor connects the actuator's piston of mass  $m_a$  to the environment, represented by a mass  $m_e$ , stiffness  $k_e$  and damping  $d_e$ . Further,  $A_i$  and  $A_o$  represent the effective inner and outer areas of the piston,  $\tau$  and  $k_{sp}$  are gains describing the valve dynamics, while  $K_s$  and  $K_p$  are load and pressure-dependent variables, respectively. Finally, parameter  $C$  is a constant arising from the linearisation procedure around a specified operating point, defined as:

$$C = \frac{1}{\beta} \left( \frac{\bar{V}_i + \bar{V}_o}{2} \right)$$

where  $\bar{V}_i$  and  $\bar{V}_o$  represent the initial volumes of hydraulic fluid trapped in the blind and the rod sides of the piston and  $\beta$  is the effective bulk modulus of the fluid.

A list of all parameters defining the linearised transfer function is given in Table 4. For each parameter a minimum, maximum and nominal value is given. The parameters are assumed to vary independently between their corresponding extreme values.

Parameter	Nominal value	Range
$k_e$	75 (KN/m)	50 – 100
$K_s$	0.375 ( $m^3/pa.s$ )	0.25 – 0.5
$K_p$	$2.5 \times 10^{-12}$ ( $m^2/s$ )	$0 - 5 \times 10^{-12}$
$C$	$1.5 \times 10^{-11}$ ( $m^3/pa$ )	$1 \times 10^{-11} - 3 \times 10^{-11}$
$d$	700 (N/m/s)	600 – 800
$m_a$	20 (Kg)	19.9 – 20.1
$A_i$	0.00203 ( $m^2$ )	0.00193 – 0.00213
$A_o$	0.00152 ( $m^2$ )	0.00144 – 0.00160
$k_{sp}$	0.0012 (m/V)	0.0011 – 0.0013
$\tau$	35 (ms)	30 – 40

**Table 4: Operating values and parameter ranges**

The uncertainty in  $K_s$  and  $K_p$  reflects variations in the operating point (especially the non-linearity arising at the interface between positive and negative spool displacements), supply pressure and orifice area gradient. Uncertainty in parameters  $k_e$  and  $d$  model variations in environmental stiffness and damping, while uncertainty in valve characteristics is modelled by variation in parameters  $\tau$  and  $k_{sp}$  [40]. Variation in parameter  $C$  reflects changes in the fluid bulk modules and the volumes of the fluid trapped at the sides of the actuator. All these parameters are known to affect the dynamic stability of the system.

Following [40], the design frequencies were chosen as  $\Omega = \{0.01, 0.05, 0.1, 0.5, 15, 10, 50, 70, 100\}$  rads/s. The robust tracking bounds are defined by the magnitude frequency response of the two systems:

$$B_u(s) = \frac{\frac{s}{2.8} + 1}{\left(\frac{s}{4} + 1\right) \left(\frac{s}{7} + 1\right) \left(\frac{s}{8} + 1\right)}$$

and

$$B_l(s) = \frac{1}{\left(\frac{s}{4.8} + 1\right) \left(\frac{s}{80} + 1\right) \left(\frac{s^2}{50} + \frac{9.6s}{50} + 1\right)}$$

These were obtained by step-response figures of merit related to rise-time, percent overshoot and settling time [40]. Thus the design specifications for the feedback controller  $K(s)$  are:

$$\max_{p \in \mathcal{P}} \left| \frac{G(p, j\omega_i) K(j\omega_i)}{1 + G(p, j\omega_i) K(j\omega_i)} \right|_{dB} \leq |B_u(j\omega_i)|_{dB} - |B_l(j\omega_i)|_{dB}$$

which should hold for all 10 design frequencies in  $\Omega$  and for every possible combination of the ten uncertain parameters varying over their respective ranges specified in Table 4. The frequency response of the tracking specification bounds  $B_u(j\omega)$  and  $B_l(j\omega)$  are displayed in Figure below (design frequencies are indicated by circles).

It is further required that the open-loop frequency response (for any permissible combination of parameters) should not enter the  $M = 1.4$  circle, which gives the design an approximate gain margin of 3 dB. Finally, since for hydraulic actuators of this type the valve dead-band typically produces a steady-state-error in the system response [40], integral action is required from the feedback controller to eliminate the steady-state error.



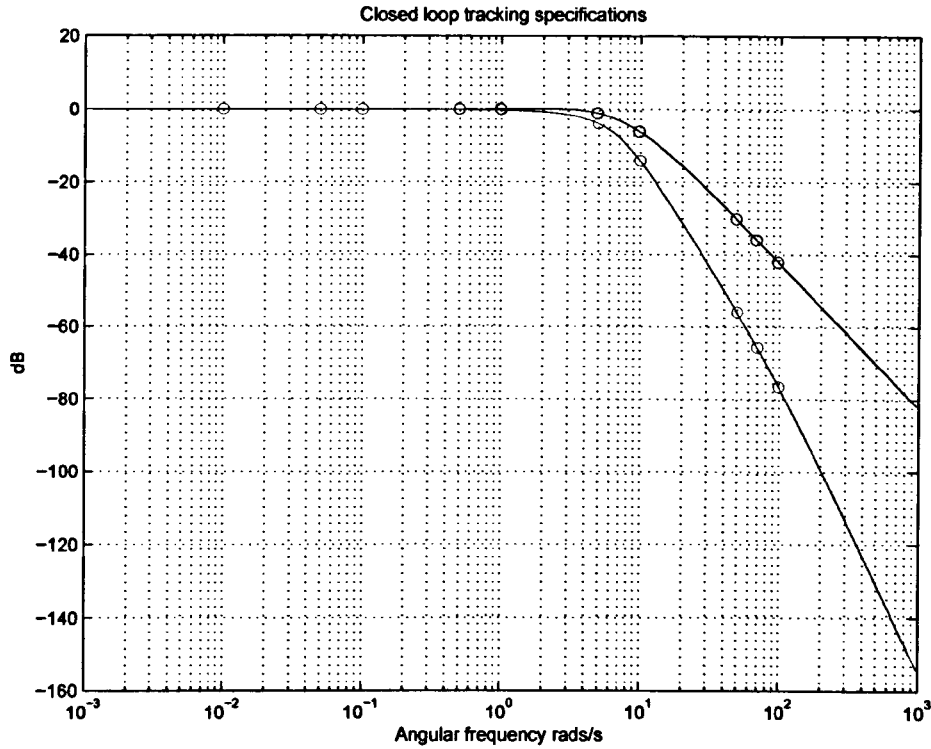


Figure 7.3: Closed-loop design tracking specifications, upper and lower bounds

The uncertainty templates of the model were first obtained by using a three-point grid for each uncertain parameter (nominal, minimum and maximum value). This resulted in  $3^{10} = 59049$  uncertain points in the Nichols chart for each template. To reduce the number of subsequent calculations, the convex hull of each uncertainty template was also obtained and used to derive the Horowitz bounds at each design frequency; this process introduces some measure of conservativeness to the design, as the uncertainty templates need not be convex, which in this case, however, is minimal. Three uncertainty templates of the plant at the fifth, sixth and seventh design frequencies (along with their convex hull) are shown in Figures 7.4, 7.5, 7.6.

The Horowitz bounds were next calculated numerically at the ten design frequencies, with a gain tolerance of 0.1 dB and a phase step of  $1^\circ$ . This was followed by the construction of the  $\mathcal{U}$ -contour, corresponding to an  $M$ -circle with  $M = 1.4$  and a high-frequency uncertainty spread of  $V_\infty = 11.03$  dB, calculated analytically from the model. The corresponding contours are shown in Figure 7.7, together with the nominal frequency response of the plant (the ten design frequencies being marked with a circle). Note that only the first seven design frequencies correspond to open Horowitz

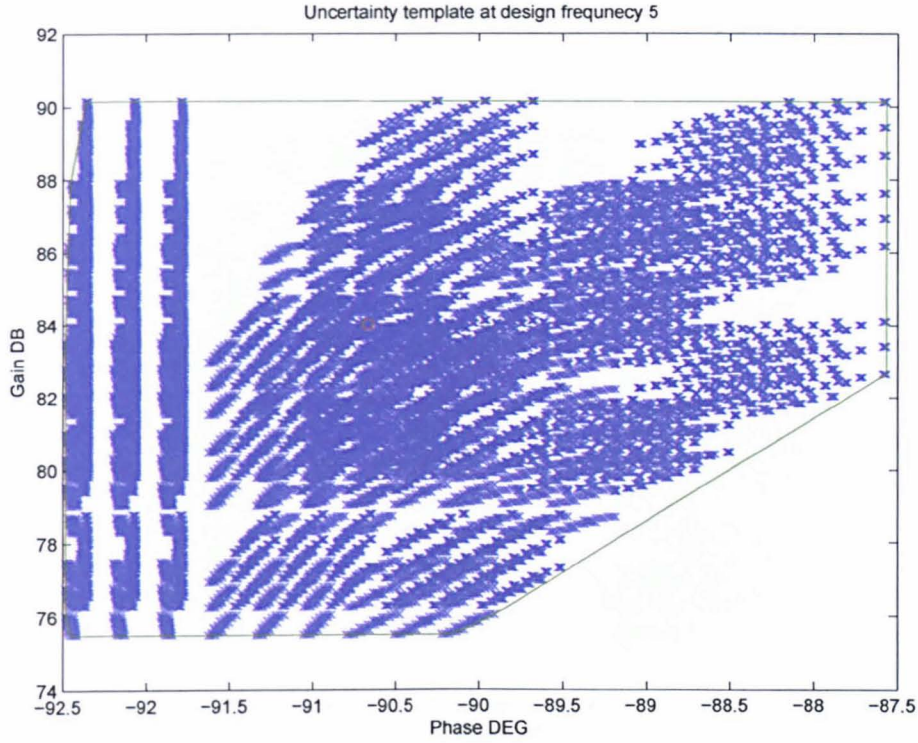


Figure 7.4: Uncertainty template, fifth design frequency

templates.

The design specifications indicate that integral action must be introduced via the feedback controller. First, it was attempted to design an optimal PID controller using the results of Theorem 1, which proved to be infeasible. The reason is clear from Figure 7.7 which indicates that a large amount of phase advance (exceeding  $90^\circ$ ) should be introduced in the mid-high frequency range. Thus the controller structure was modified as:

$$K_1(s) = \frac{k_1 + k_2 s + k_3 s^2}{s \left( \frac{s}{130} + 1 \right)}$$

The  $s$ -term in the denominator provides the required integral action, while the numerator is a  $PDD^2$  (proportional-derivative-double-derivative) term providing sufficient phase-advance (up to  $180^\circ$  at high frequencies). The additional pole at  $s = -130$  was introduced to ensure that the controller is proper. Next, the denominator term of  $K_1(s)$  was absorbed to the nominal plant, and the three-parameters  $k_1$ ,  $k_2$  and  $k_3$  were optimized using the algorithm of section 3 and the results of Theorem 1'. The cost-function chosen for optimization was the open-loop asymptotic gain (controlled by  $k_3$ ). The new optimisation problem proved feasible and resulted in an optimal

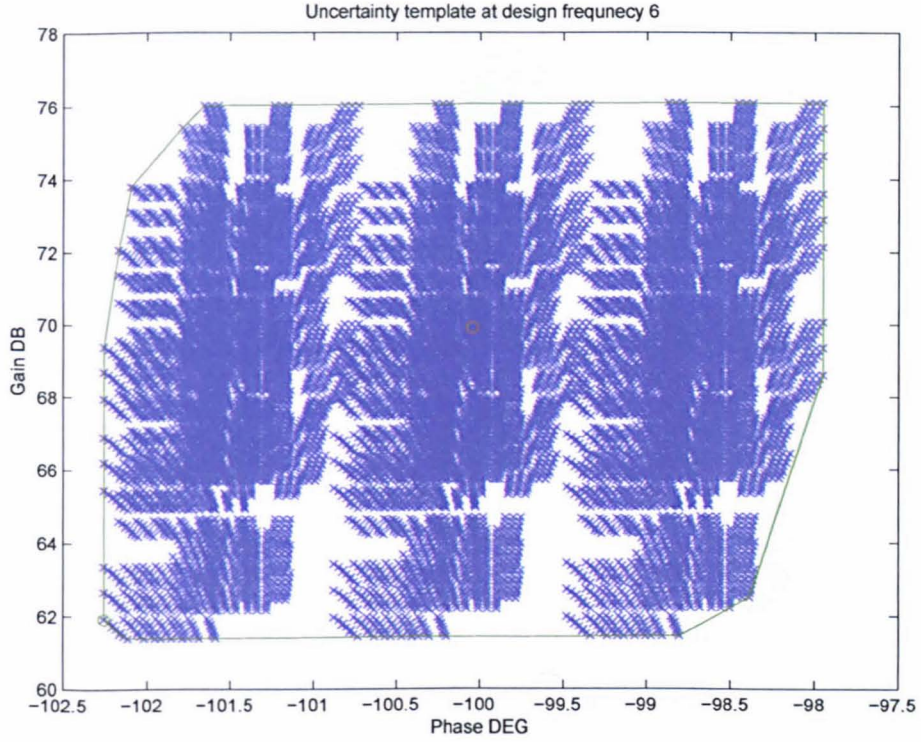


Figure 7.5: Uncertainty template, sixth design frequency

$PDD^2$ -controller with

$$k_1^* = 0.004, \quad k_2^* = 0.002 \quad k_3^* = 4.9778 \times 10^{-5}$$

The resulting open-loop system is shown in Figure 7.8.

It can be seen that the design specifications at all ten design frequencies are satisfied, the first seven points lying on or above the corresponding templates, the last three (high frequencies corresponding to the closed Horowitz contours) lying outside or on the  $\mathcal{U}$ -template. It may be seen, however that the nominal frequency response penetrates the  $\mathcal{U}$ -contour between the two consecutive design frequencies  $\omega = 10$  and  $\omega = 50$  rad/s. This is a common problem with QFT design which is based on a discrete set of design frequencies. A typical remedy is to define a more dense set of design frequencies or tighten the specifications. Here a simpler technique was followed by adding an additional first-order lag term to the controller, to modify the open loop response in the offending frequency range  $10 \leq \omega \leq 50$  rad/s. The overall controller is:

$$K(s) = \frac{(0.004 + 0.002s + 4.9778 \times 10^{-5}s^2)(0.06231s + 1)}{s(s/130 + 1)(0.1295s + 1)}$$



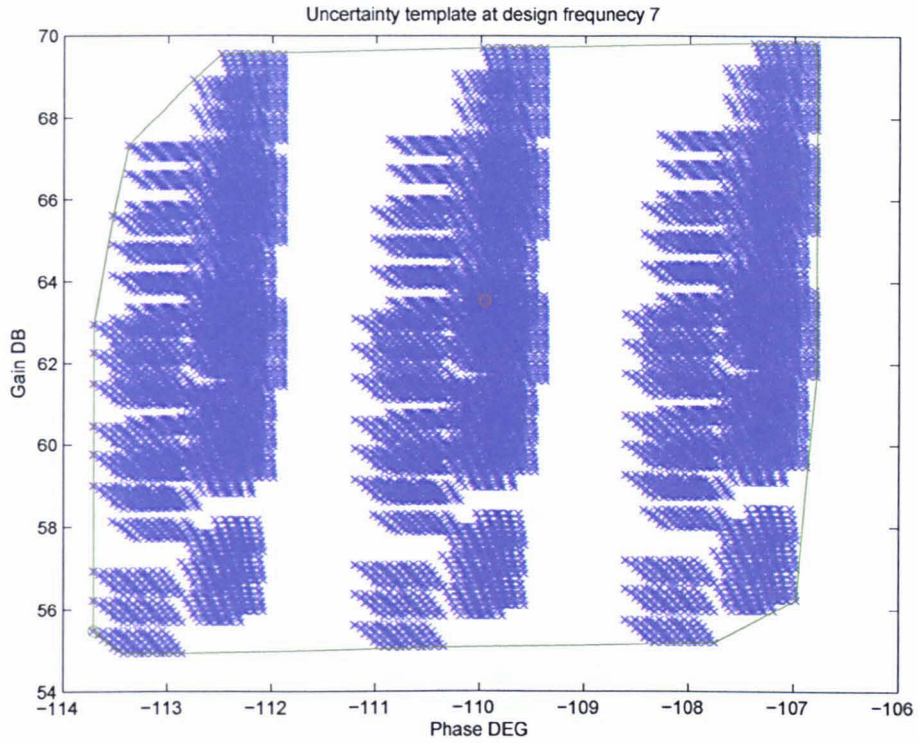


Figure 7.6: Uncertainty template, seventh design frequency

The corresponding open-loop response is shown in Figure 7.9. The nominal open-loop system has a cross-over frequency of 16.91 rads/s and 3-dB closed-loop bandwidth equal to 28.21 rads/s. The shaped open-loop response is also shown in 7.10, along with the uncertainty templates at the ten design frequencies. It may be seen that all design specifications are met at the first seven design frequencies, with the nominal open loop system lying above the corresponding performance bounds (Horowitz templates). The fact that most five of these seven points lie (almost) on the bounds is a consequence of the optimisation method that was used to design the feedback controller and indicates that the design is in agreement with the general spirit of the QFT philosophy, in the sense that the minimum possible gain is used, sufficient to meet the robust performance specifications. It can also be seen that the uncertainty templates for the last three design frequencies also lie outside the  $M = 1.4$  circle (equivalently the nominal frequency response at these frequencies lies outside the  $\mathcal{U}$ -contour, so that the robust stability objectives of the design have also been met. Again the three templates are close to the  $M$ -circle boundary which indicates that the enforced stability margins are tight (for certain combinations of the uncertain parameters). Robust stability

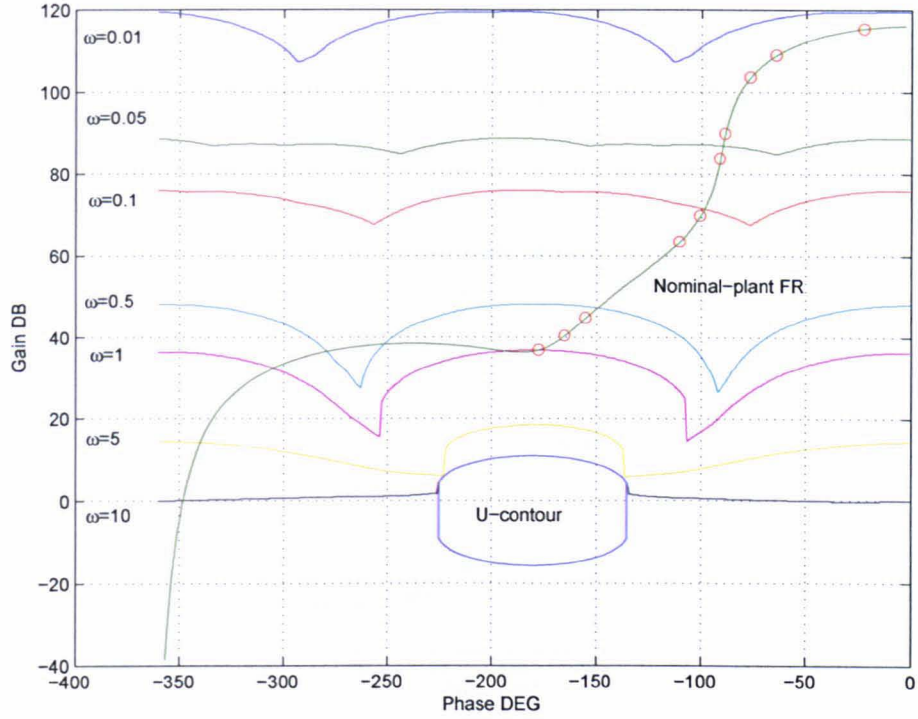


Figure 7.7: Nominal-plant frequency response, Horowitz templates and  $\mathcal{U}$ -contour

is an important consideration in this case, as the system manifests a high-frequency resonance peak. This can be clearly seen either from the Bode plots, or indirectly inferred also from the shape of the uncertainty templates, whose phase spread increases at high frequencies, as the resonance frequency shifts with parameter uncertainty.

The controller is quite similar to the one designed in [40] shown below which was validated experimentally. This suggests that the design specifications in this case are tight.

$$G(s) = \frac{0.0065(\frac{s}{4} + 1)(\frac{s}{40} + 1)}{S(\frac{s}{100} + 1)(\frac{s}{300} + 1)}$$

The last step of the QFT design is to design a pre-filter. Here the following procedure was used: First, the magnitude frequency responses of  $3^5 = 243$  open-loop uncertain systems were plotted (see 7.11). These correspond to the five more important parameters (in terms of uncertainty template spread), the remaining five parameters being fixed to their nominal value. Next, the maximum and minimum gains were recorded at the ten design frequencies. The values obtained are summarised in the table 5 below.



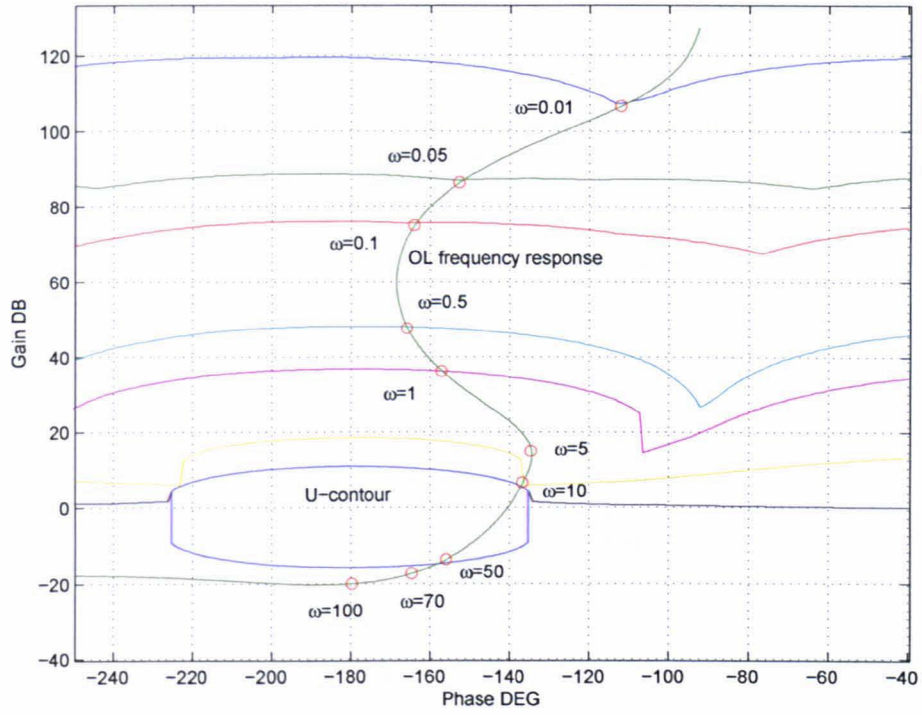


Figure 7.8: Nominal open-loop response with  $PDD^2$ -optimal controller

Frequency (rad/s)	Upper bound	Low bound	Max gain	Min gain
0.01	0.0000	0.0000	0.0000	0.0000
0.05	0.0003	-0.0004	0.0008	0.0002
0.10	0.0013	-0.0018	0.0032	0.0007
0.50	0.0300	-0.0437	0.0777	0.0177
1.0	0.1030	-0.1733	0.2796	0.0631
5.0	-1.0882	-3.8959	2.2479	0.5656
10.0	-6.1353	-14.0514	2.7065	1.3864
50.0	-30.1052	-55.7940	-4.7396	-17.7778
70.0	-35.8489	-65.5845	-8.3369	-21.1456
100.0	-41.9907	-76.4891	-7.1198	-23.9876

Table 5: Closed-loop specifications, Minimum and maximum gain (gain values in dB's)

As expected, the spread in closed-loop gain is within the required tolerances; thus all responses can be brought between the specified lower and upper bounds by designing a pre-filter which essentially provides frequency-dependent scaling. In this case, the pre-filter must provide adequate attenuation at high frequencies.

The filter was designed via an optimization procedure (linear programming). First, the difference of the maximum gain from the upper bound was recorded at the design frequencies, together with the difference of the minimum gain from the lower

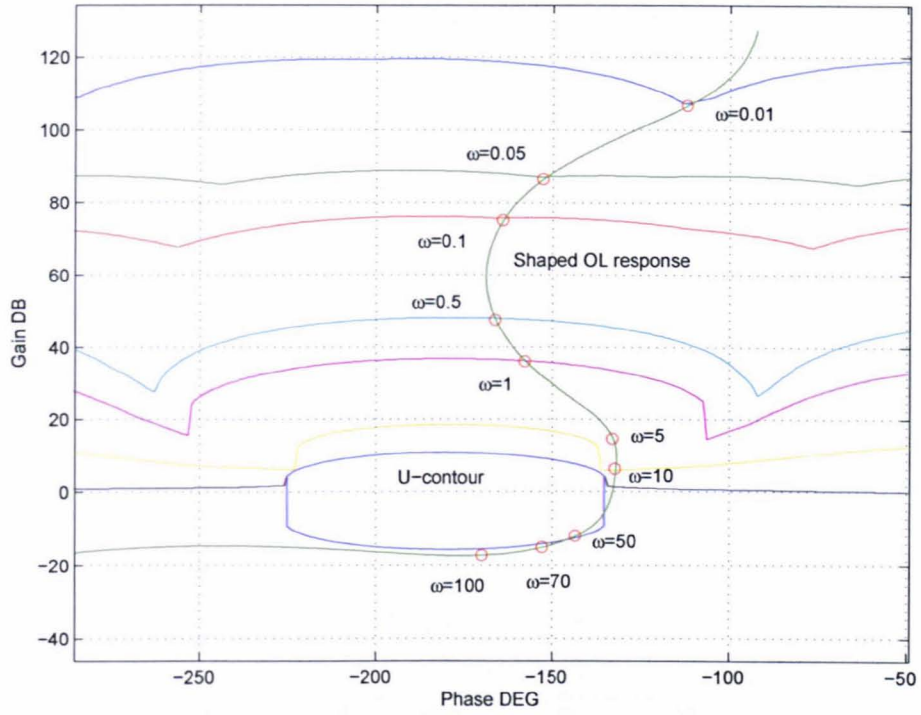


Figure 7.9: Nominal open-loop response with modified  $PDD^2$ -optimal controller

frequency bound. The two differences were then averaged and this defined the required attenuation of the pre-filter (at the ten design frequencies). This procedure was followed so as to bring the responses in the middle of the specified region in the Bode diagram. With the magnitude frequency-response of the filter specified at the ten design frequencies, the next task was to fit a stable rational function to approximate the response. Various techniques (e.g. least-squares) can be used for this purpose, but the one that was followed was based on Matlab's routine *fitmaglp.m* (from the  $\mu$ -control toolbox) in which fitting can be performed interactively over various filter orders, while the results are displayed graphically (target and achieved frequency response). The function formulates the problem as a linear programme using weights to emphasize the fit at the required frequency-ranges. In this example equal weights were used for all (ten) frequencies. A filter order equal to three was found to give a good compromise between accuracy and complexity. The transfer function of the filter was obtained as:

$$F(s) = \frac{(1 + s/16.25)(1 + s/(20.08 \pm j145.74))}{(1 + s/104.36)(1 + s/(4.52 \pm j2.09))}$$

so that  $F(s)$  is both stable and minimum-phase (as guaranteed by the routine). The specified and achieved responses of the filter at the design frequencies are summarised in the Table 6:



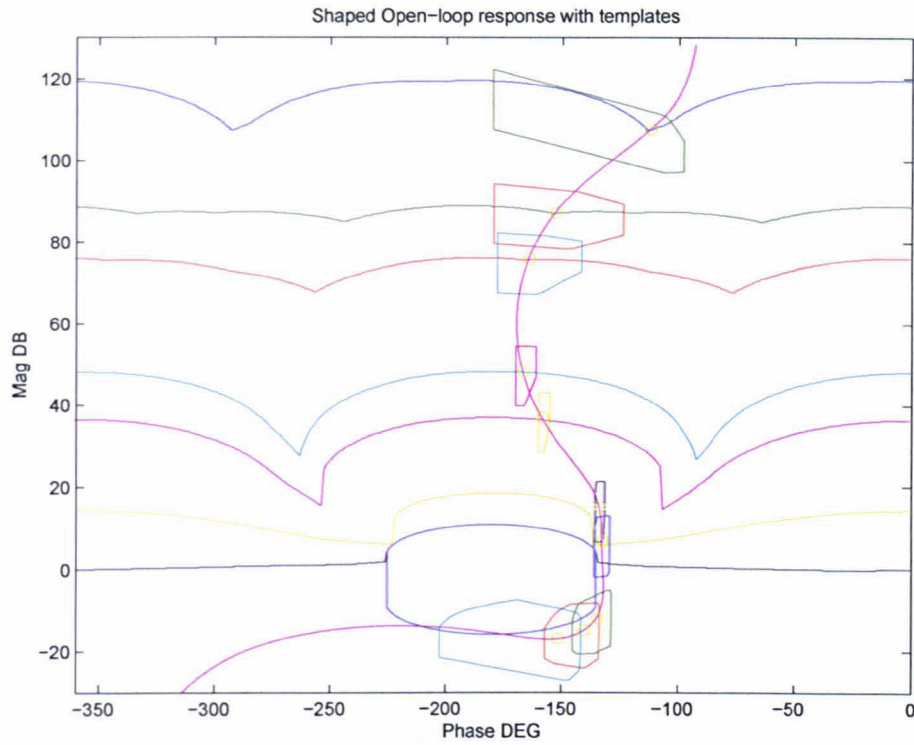


Figure 7.10: Shaped open-loop response and uncertainty templates

Frequency (rad/s)	Target gain (dB)	Achieved gain (dB)
0.01	0.0000	0.0000
0.05	-0.0005	-0.0006
0.10	-0.0021	-0.0022
0.50	-0.0528	-0.0546
1.0	-0.2120	-0.2065
5.0	-4.8403	-3.8989
10.0	-12.2030	-12.1397
50.0	-31.8440	-31.6909
70.0	-36.7536	-35.9755
10.00	-43.9554	-43.6862

Table 6: Target and achieved pre-filter gains

The closed-loop responses of the system (with pre-filter) are shown in Figure 7.12. As expected, these are all contained within the specified upper and lower bounds. The  $3^5$  (unit) step responses of the system are finally shown in Figure 7.13.



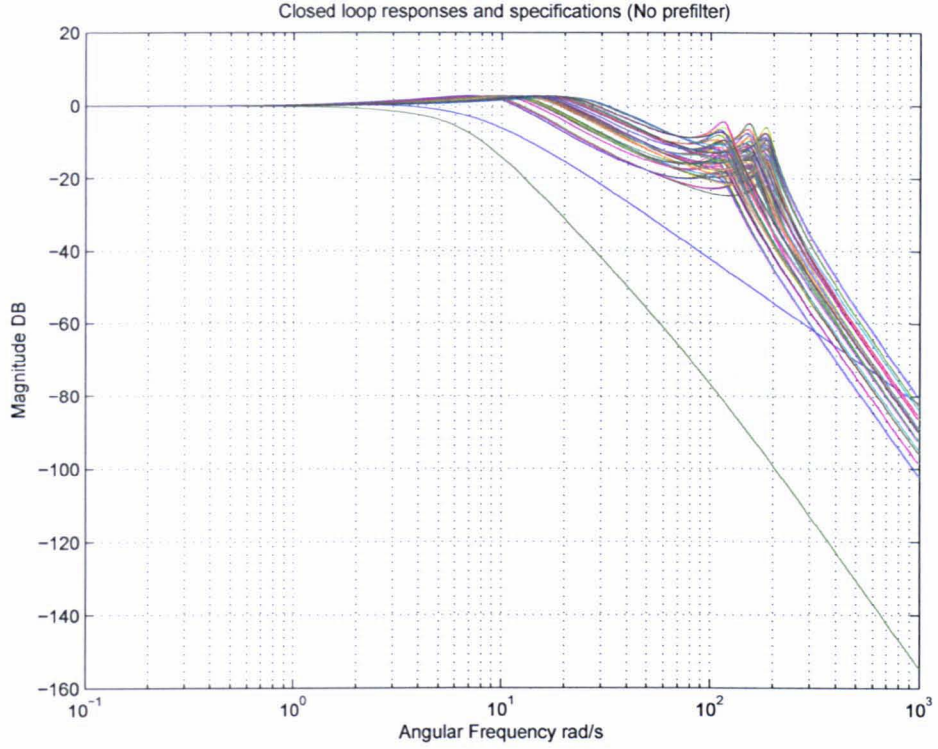


Figure 7.11: Closed-loop frequency responses - No pre-filter

## 7.5 Conclusions

The chapter has carried out a design of a hydraulic actuator model interacting with an uncertain environment using the QFT design methodology. First, the model of the actuator has been derived from first principles using a linearisation procedure and a number of simplifying assumptions. This procedure has quantified the uncertainty of the model's parameters and this information has been used in the context of the design procedure to derive the robust stability and performance objectives, which were then translated into appropriate Nichol's plots bounds. The optimization-based loop shaping techniques developed in earlier chapters have proved successful in designing a low-degree robust QFT controller, despite the fact that the chosen specifications are rather tight. The design procedure is systematic, fast and almost completely automated, although a slight modification of the feedback controller was needed at the last step of the design to account for the fact that a small number of design frequencies had been chosen. A pre-filter has been designed using a systematic procedure using optimization (linear programming). The effectiveness of the design was illustrated with numerous simulations.

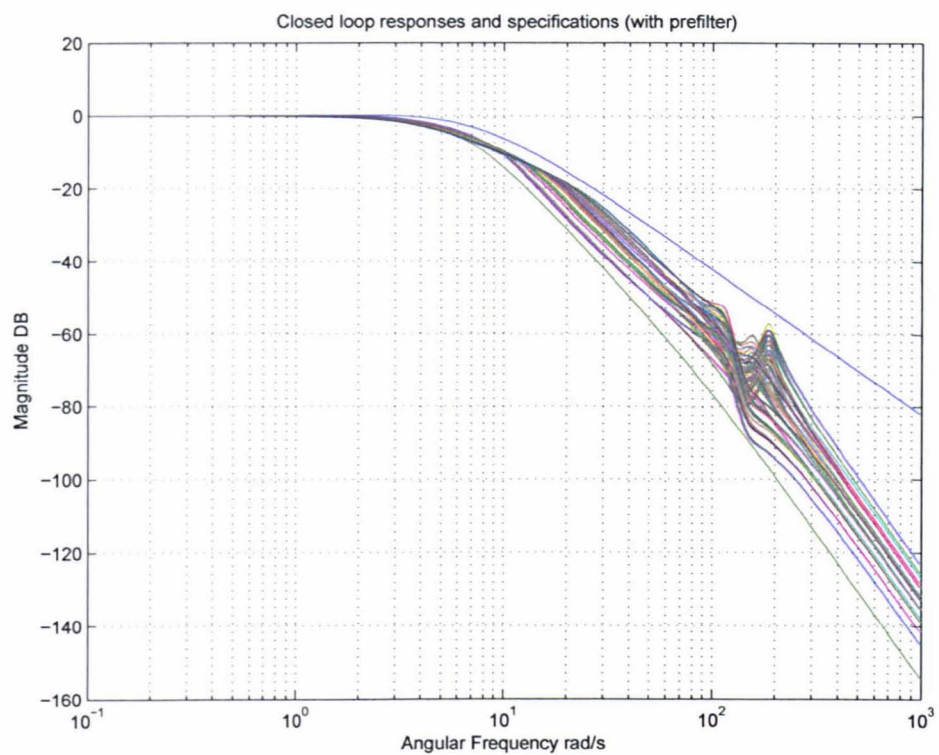


Figure 7.12: Closed-loop frequency responses - With pre-filter

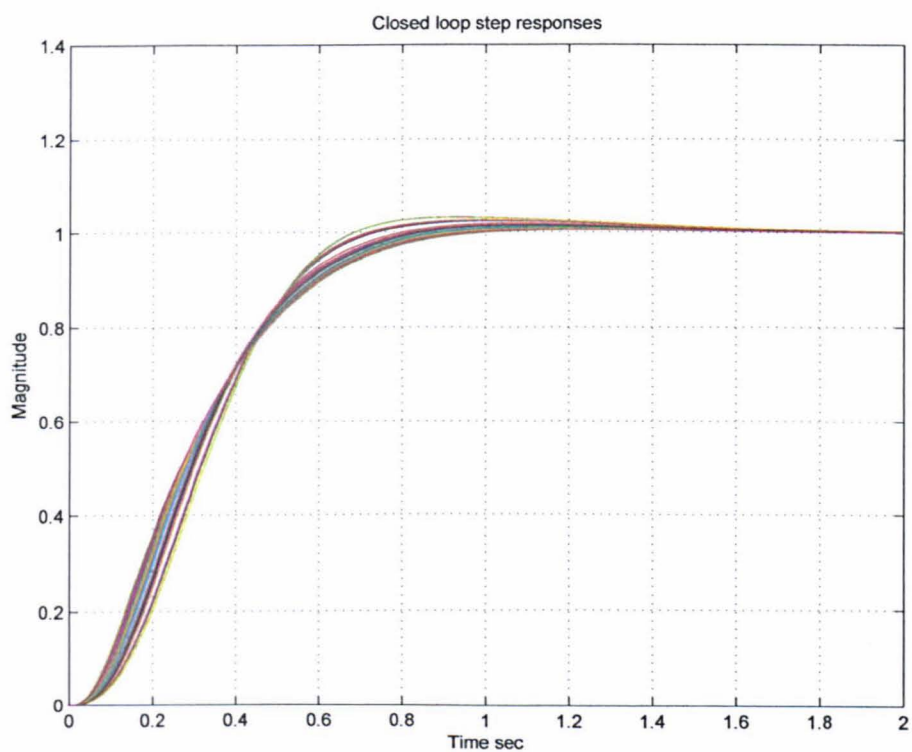


Figure 7.13: Closed-loop step responses - With pre-filter

# Chapter 8

## Conclusions, Future Work

### 8.1 Conclusions

The thesis is in the area of robust control design using the QFT methodology. The introductory chapters have outlined the main steps of the design procedure corresponding to this method, the techniques for implementing each step and aspects of background theory including recent developments.

A significant amount of effort has been dedicated to the derivation of optimisation algorithms for designing robust controllers within the QFT framework. The main objective here was to automate the control-design step of the procedure which is the most complex aspect of the method. Two main approaches have been followed:

- *Design of fixed structure controllers:* The proposed algorithms automate the loop-shaping step of the QFT design procedure. They can be applied to design robust-performance optimal controllers of a simple structure (PID, phase lead/lag, second order) which are widely used in industrial control. Moreover, the method can be used for designing more complex controllers involving arbitrary interconnections of these optimised structures. The main algorithm relies on the fact that fixing the phase of a simple-structure controller at two distinct frequencies, fixes its phase over *all* frequencies and thus essentially determines its dynamic part. Thus, gain optimisation can be performed to provide an optimal design which minimises an appropriate criterion (asymptotic open-loop gain, nominal or worst-case bandwidth, nominal or worst-case cross-over frequency etc), while satisfying the robust stability and performance constraints imposed by the specifications. A search over a two-dimensional phase grid can then be

performed to produce the overall optimal design. A number of design examples have demonstrated the validity of this method in producing effective designs.

- *Design of optimal controllers in the frequency domain:* This method is more ambitious. Instead of fixing the structure of the controller and optimising its parameters, the method attempts to perform the optimisation directly in the frequency domain. The robust stability and performance constraints are formulated as linear inequalities by approximating locally the Horowitz templates (robust performance templates) and the  $\mathcal{U}$ -contour (robust stability contour). Consistently with the QFT design philosophy, over-designing the system is avoided by minimising the asymptotic open-loop gain of the system. The main difficulty arising from this approach is to ensure that the solution actually corresponds to the frequency-response of a realisable LTI system, which can be implemented as the feedback controller. To ensure that this can always be done, two additional sets of linear constraints are imposed to the Linear Programming problem. The first set is obtained by discretising the Bode gain/phase integral relationship and imposes “realisability” constraints to the optimisation problem; similarly, the second set imposes “analyticity” constraints to the optimisation. The two additional sets of constraints now ensure that a realisable and smooth LTI controller can be recovered by approximating the optimised frequency response.

Since QFT is essentially a CAD design approach, a significant part of the work involved the development of a Matlab software tool which implements the main steps of the method, i.e. generation of uncertainty templates on the Nichol’s chart, calculation of robust stability and performance bounds (Horowitz templates,  $\mathcal{U}$ -contour), manual or optimised loop-shaping and pre-filter design, etc.

A detailed case study using the developed methods has been carried out in the last part of the thesis. This involves the design of a non-linear hydraulic actuator interacting with its environment. The non-linear model was linearised around an operating point resulting in a nominal model depending on ten uncertain parameters. An optimal loop-shaping design procedure was applied to this model resulting in a feedback QFT controller and a prefilter which were validated via extensive simulations. The resulting design successfully met the robust stability and performance specifications, indicating

that the method can be applied successfully to uncertain systems of at least medium complexity.

## 8.2 Limitations

The main limitation of the work is that it is restricted to the control design of SISO systems. Although most methods presented in the thesis can be extended to the multivariable case using a standard approach involving a diagonal feedback controller, this would increase considerably the complexity of the software tool and the optimisation problems used for automatic loop-shaping. Throughout the work only finite-dimensional LTI systems have been considered. Extensions to models with uncertain time delays are straight-forward, in contrast to time-varying or non-linear models, for which the design methodology, although in principle applicable, is in practice much more demanding. Note however, that most non-linear systems are normally tackled through linearization around an operating point, similar to the procedure followed in the case-study.

## 8.3 Future Work

The following issues extending the results of this research can be investigated in future work:

- Potential improvements of the optimisation algorithms presented in the thesis can be investigated, e.g. the formulation of non-linear optimisation problems which do not require the approximation of the robust stability and performance constraints.
- Frequency-domain approximation algorithms can be developed for obtaining the transfer function of the feedback controller from its frequency response (when the second optimisation approach is used), and for the automatic derivation of the pre-filter when tracking specifications are included in the design. Complexity issues can also be addressed and tackled via controller model-reduction techniques which quantify performance deterioration and can thus serve as transparent indicators of the trade-off between controller complexity and optimality.

- The derivation of novel robust-control design methods can be attempted by exploring the synergy between QFT and  $\mathcal{H}_\infty$  optimal control methodologies. The idea here is to combine the relative merits of the two techniques in a single powerful design methodology. A possible approach involves the formulation of QFT bounds in terms of “structured singular value” inequalities which can be solved via known techniques (e.g. Linear Matrix Inequalities).
- Extension of the developed optimisation techniques to the multivariable case. At present matrix problems are normally tackled in the QFT framework by designing diagonal feedback controllers and decomposing the design to a number of (uncoupled) MISO (Multiple Input Single Output) problems. The systematic extension of the design methodology to feedback controllers with non-diagonal structure is still an open research issue.
- Development of a user-friendly menu-driven CAD environment in which QFT designs can be performed and evaluated.

# Bibliography

- [1] Miani. Stefano Blanchini. Franco, Lepschy. Antonio and Viaro. Umberto. Characterization of pid and lead/lag compensators satisfying given specifications. *IEEE transactions on automatic control ISSN 0018-9286 CODEN IETAA9*, 49(5):736–740, 2004.
- [2] H.W. Bode. Network analysis and feedback amplifier design. *Van Nostrand*, 1945.
- [3] Chait. Yossi Borghesani. Craig and Yaniv. Oded. The qft frequency domain control design toolbox for use with matalb.
- [4] Y. Chait Bounds, J. M. Rogues and C.V. Hollo. A new algorithm for computing qft. *Proceedings of the 36th Conference on Decision and Control, San Diego, U.S.A.*, Dec 1997.
- [5] M. Breiner. Synthesis of control system ffor a single-input multi-output plant. *International Journal of Control*, 12, no.9:1114–1124, 1981.
- [6] G. F. Bryant and G. D. Halikias. Optimum loop-shaping for systems with large parameter uncertainty via linear programming. *International Journal of Control ISSN 0020-7179 CODEN IJCOAZ*, 62(3):557–568, 1995.
- [7] L. Richard. Burden and J. Douglas. Faires. *Numerical Analysis*, volume 0534382169. Brooks/Cole, 2001.
- [8] Richard C.Dorf. *Modern Control Systems*, volume Sixth Edition. Addison-Wesley, 1992.
- [9] Y. Chait. Qft loop shaping and minimisation of the high-frequency gain via convex optimisation. *Symposium on Quantitative Feedback Theiry and Other Frequency Domain Methods and Applications.*, pages 13–27, 1997.

- [10] Chen. Qian Chait. Yossi and Hollot.C.V. Automatic loop-shaping of qft controllers via linear programming. *Journal of dynamic systems, measurement, and control* ISSN 0022-0434 CODEN JDSMAA, 121(3):351–357, 1999.
- [11] Yeong-Hwa Chang and Jin-Chuan Chang. Robust controller design of multivariable perturbed systems with qft. *American Control Conference, 1999. Proceedings of the 1999*, 2:1405 – 1409, 1999.
- [12] W. Chen and D. J. Ballance. Stability analysis on the Nichols chart and its application in QFT. *Control 1991. ,International Conference onControl '91.*, 1:476–480, August 1991.
- [13] W. Chen and D. J. Ballance. QFT design for uncertain non-minimum phase and unstable plants. *Preprint, University of Glasgow*, July 1997.
- [14] Godoy. Emmanuel Croulard. Vincent and Boichot. Jtrome. Qft controller optimization for automatic design. *Proceedings of the 39th IEEE CDC*, December 2000.
- [15] B. George. Dantzig. *Linear Programming and Extensions*. Princeton, 1972.
- [16] J. D'Azzo and C. Houpis. *Linear Control Systems Analysis And Design*. McGraw-Hill, 1995.
- [17] J. D'Azzo and C. Houpis. *Feedback control systems analysis and synthesis*. Prentice Hall, 1998.
- [18] A. Bruce. Doyle, C. John. Francis and Tannenbaum. R. Allen. *Feedback control Theory*. Macmillan, 1992.
- [19] S. J. Freudenberg and P. D. Looze. *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1988.
- [20] I. Egana Garcia-Sanz, M. and M. Barreras. Design of quantitative feedback theory non-diagonal controllers for use in uncertain multiple-input multiple-output systems. *IEE Proceedings-Control Theory and Applications*, 152(2):177–187, 2005.
- [21] M. Garcia-Sanz and I. Eguinoa. Improved non-diagonal mimo qft design technique considering non-minimum phase aspects. *7th Int. Symp. on QFT and Robust Frequency Domain Methods, Lawrence, Kansas, USA.*, pages 177–187, 2005.



- [22] Sardar Gautam and P. S. V. Nataraj. A template generation algorithm for non-rational transfer functions in qft designs. *Proceedings of the 36th Conference on Decision and Control, San Diego, U.S.A.*, Dec 1997.
- [23] Abbas Emami-Naeini Gene F. Franklin, J. David Powell. *Feedback Control of Dynamic Systems*, volume Third Edition. Addison-Wesley, 1994.
- [24] A. Gera and I. Horowitz. Optimisation of the loop transfer function. *International Journal of Control*, 31:389–398, 1980.
- [25] Michael. Green and N. J. David. Limebeer. *Linear Robust Control*. Prentice Hall, 1996.
- [26] P.O. Gutman. Robust certainty equivalence - a new principle for adaptive control. *IFAC Workshop on Robust Adaptive Control*, 1988.
- [27] Nandakumar.R. Halikias.G.D and A.C. Zolotas. A new educational software tool for robust control design using the qft method. *IEEE International conference on Decision and Control*, Dec 2003.
- [28] I. M. Horowitz. *Synthesis of Linear Systems*. Academic Press, 1973.
- [29] I. M. Horowitz and M. Sidi. Synthesis of linear systems with large plant ignorance for prescribed time-domain tolerances. *Int. J. Control*, 16:287–309, 1972.
- [30] I. M. Horowitz and M. Sidi. Optimum synthesis of non-minimum phase systems with plant uncertainty. *Int. J. Control*, 27:361–386, 1978.
- [31] Ballance. D. J and Gawthrop. P. J. Control systems design via a quantitative feedback theory approach.
- [32] S. Jayasuriya and Y. Zhao. Robust stability of plants with mixed uncertainty and quantitative feedback theory. *Proceedings of the 1993 American Control Conference.*, 2:1706–1710, 1993.
- [33] S. Jayasuriya and Y. Zhao. Stability quantitative feedback designs and existence of robust qft controllers. *International Journal of Robust and Nonlinear Control*, 4, no.1:21–46, 1994.

- [34] Suhada.Jayasuriya Ken. Murray and Asokanthad. Samuel. Robust stability in sequential mimo qft. *American Control Conference, 2003. Proceedings of the 2003*, 6:4834– 4839, 2003.
- [35] Keunsik Kim. and Young Chol Kim. The complete set of pid controllers with guaranteed gain and phase margins.
- [36] M. L. Kerr Lan, C. and S. Jayasuriya. Synthesis of controllers for non-minimum phase and unstable systems using non-sequential mimo quantitative feedback theory. *American Control Conference, Boston, MA*, pages 219–225, 2004.
- [37] Y. Chait Lee, J.W and Steinbuch. Maarten. On qft tuning of multivariable mu controllers. Dec.
- [38] J. M. Maciejowski. *Multivariable feedback design*. Addison-Wesley, 1989.
- [39] Kerr. Murray and S Jayasuriya. Sufficient conditions for robust stability in non-sequential mimo qft. *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, 2:1813– 1817, December 2003.
- [40] Sepehri Nariman. and Niksefat Navid. Robust force controller design for an electro-hydraulic actuator based on nonlinear model. *Proceedings of the 1999 IEEE ICRA*, 1:0272–1708, 1999.
- [41] Sepehri Nariman. and Niksefat Navid. Designing robust force control of hydraulic actuators. *IEEE Control Systems Magazine*, 1:0272–1708, 2001.
- [42] Nataraj.P.S.V. A matlab toolbox for qft-based synthesis of linear/nonlinear lumped and linear distrubed systems.
- [43] O.Yaniv N.Cohen, Y.Chait and C.Borghesani. Stability analysis using nichols charts. *Proceedings of the Symposium on Quantitative Feedback Theory*, pages 80–103, 1992.
- [44] Sidi.M. Feedback synthesis with plant ignorance, non-minimum phase, and times-domain tolerances. *Automatica*, 265-271 1976.
- [45] Sidi.M. Maximisation of gain-bandwidth in sampled systems. *International Journal of Control*, 32:1099–1109, 1980.

- [46] Sigurd. Skogestad and Ian. Postlethwaite. *Multivariable Feedback Control*. John Wiley and sons, 1996.
- [47] J. Stoer and R.Bulirsch. *Introduction to Numerical Analysis*, volume Second Edition. Springer-Verlag, 1992.
- [48] Y. Theodor and U. Shaked.  $\mathcal{H}_\infty$  multiple objective robust controllers for infinite-horizon single measurment single control input problems. *IEEE Transactions on Automatic Control*, 39, no.10:2130–2134, 1994.
- [49] David F. Thompson. Gradient formulations for sensitivity-based qft performance bounds.
- [50] David F. Thompson and John S. Pruyne. On characterizing sensitivity-based and traditional formulations for quantitative feedback theory. *INT. J. CONTROL*, 72(6):536–545, 1999.
- [51] D.F. Thompson and O.D.I. Nwokah. Analytic loop shaping methods in quantitative feedback theory,. *Transaction of the ASME, Journal of Dynamic Systems, Measurment and Control*, 116,no.2:169–177, 1994.
- [52] Dutton.K Thompson.S and Barraclough.B. *The Art of Control Engineering*. Addison-Wesley, 1997.
- [53] Ukpai I. Ukpai and S. Jayasuriya. Gain-bandwidth optimization of pid controllers. *Proceedings of the American Control Conference ISSN 0743-1619*, 6:4979–4984, 2002.
- [54] P.A. Unstead and I.M. MacLeod. Synthesis of continuous-time minimum-phase frequency response specifications. *Pre-Print*, 1999.
- [55] Hwang C.Y. Wang, B. and T.P. Tasi. Synthesis of feedback control systems with large plant uncertainty and time-varying gain, feedback systems with plant uncertainty ,. *International Journal of Control*, 51, no3:687–704, 1990.
- [56] Gutman P.O. Yaniv, O. and L. Neumann. An algorithm for adaotation of a robust controller to reduced plant uncertainty. *Automatica*, 26, no.4:709–720, 1990.

- [57] O. Yaniv and I. M. Horowitz. A quantitative design method for MIMO linear feedback systems having uncertain plants. *Int. J. Control*, 43:401–421, 1986.
- [58] Oded Yaniv. *Quantitative Feedback Design of Linear and Nonlinear Control Systems*. Kluwer Academic Publishers, 1999.
- [59] Odesand. Yaniv and Nugurka. Mark. Automatic loop shaping of low order controllers. *Proceedings. 2004 23rd IEEE Convention of Electrical and Electronics Engineers in Israel, 2004.*, pages 29–32, September 2004.
- [60] Odesand. Yaniv and Nugurka. Mark. Automatic loop shaping of structured controllers satisfying qft performance. *Journal of dynamic systems, measurement, and control ISSN 0022-0434 CODEN JDSMAA*, 127(3):472–477, 2005.
- [61] Y.Chait and O.Yaniv. Direct control design in sampled-data uncertain systems. *Automatica*, 29:365–372, 1993.
- [62] Kemin. Zhou and John C. Doyle. *Essentials of Robust Control*. Prentice-Hall, 1998.
- [63] A. C. Zolotas and G. D. Halikias. Optimal design of PID controllers using the QFT method. *IEE Proceedings - Control Theory Appl.*, 146(6), November 1999.

# Chapter 9

## Appendix

### 9.1 Appendix A

**Derivation: Bode Approximation:**

Since

$$\lambda_i = \ln \left[ \frac{\zeta}{\omega_i} \right] = \ln \left[ \frac{\omega_j}{\omega_k} \right]$$

the approximated Bode integral equation can be written as

$$\begin{aligned} \phi(\omega_k) &= \frac{1}{\pi} \int_{-\infty}^{\ln \frac{\omega_1}{\omega_k}} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \\ &+ \frac{1}{\pi} \sum_{j=1}^{n-1} \int_{\ln \frac{\omega_j}{\omega_k}}^{\ln \frac{\omega_{j+1}}{\omega_k}} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \\ &+ \frac{1}{\pi} \int_{\ln \frac{\omega_n}{\omega_k}}^{\infty} \frac{d \ln |L(j\zeta)|}{d\lambda} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \end{aligned}$$

By bringing the constant terms outside the integral we get

$$\phi(\omega_k) \cong \frac{1}{\pi} \frac{\ln \left( \frac{L_{j+1}}{L_j} \right)}{\ln \left( \frac{\omega_{j+1}}{\omega_j} \right)} \int_{-\infty}^{\ln \frac{\omega_1}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.1)$$

$$+ \frac{1}{\pi} \sum_{j=1}^{n-1} \frac{\ln \left( \frac{L_n}{L_{n-1}} \right)}{\ln \left( \frac{\omega_n}{\omega_{n-1}} \right)} \int_{\ln \frac{\omega_j}{\omega_k}}^{\ln \frac{\omega_{j+1}}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.2)$$

$$+ \frac{1}{\pi} \frac{\ln \left( \frac{L_2}{L_1} \right)}{\ln \left( \frac{\omega_2}{\omega_1} \right)} \int_{\ln \frac{\omega_n}{\omega_k}}^{\infty} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.3)$$

$$(9.4)$$

Now say

$$A_{nk} = \frac{1}{\pi} \frac{1}{\ln\left(\frac{\omega_{j+1}}{\omega_j}\right)} \int_{-\infty}^{\ln \frac{\omega_1}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.5)$$

and

$$A_{ok} = \frac{1}{\pi} \frac{1}{\ln\left(\frac{\omega_n}{\omega_{n-1}}\right)} \int_{\ln \frac{\omega_j}{\omega_k}}^{\ln \frac{\omega_{j+1}}{\omega_k}} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.6)$$

and

$$A_{jk} = \frac{1}{\pi} \frac{1}{\ln\left(\frac{\omega_2}{\omega_1}\right)} \int_{\ln \frac{\omega_n}{\omega_k}}^{\infty} \ln \coth \left| \frac{\lambda}{2} \right| d\lambda \quad (9.7)$$

where  $j = 1, 2, \dots, n-1$ .

Therefore by rewriting equation's 7.1, 7.2, 7.3 based on equation's 7.5, 7.6, 7.7 we get

$$\begin{aligned} \phi_k &= \sum_{j=1}^{n-1} A_{jk} [\ln(L_{j+1}) - \ln(L_j)] \\ &\quad + A_{nk} [\ln(L_n) - \ln(L_{n-1})] \\ &\quad + A_{ok} [\ln(L_2) - \ln(L_1)] \end{aligned}$$

where  $k = 1, 2, \dots, n-1$

Expanding the above equation we get

$$\begin{aligned} \phi_k &= A_{1k} [\ln(L_2) - \ln(L_1)] + A_{2k} [\ln(L_3) - \ln(L_2)] \\ &\quad + \dots + \\ &\quad + A_{(n-1)k} [\ln(L_n) - \ln(L_{n-1})] \\ &\quad + A_{nk} [\ln(L_n) - \ln(L_{n-1})] \\ &\quad + A_{ok} [\ln(L_2) - \ln(L_1)] \end{aligned}$$

simplifying the above equation we get

$$\begin{aligned}
\phi_k &= (-A_{0k} - A_{1k}) \ln(L_1) + (A_{0k} + A_{1k} - A_{2k}) \ln(L_2) + (A_{2k} - A_{3k}) \ln(L_3) \\
&+ ..... \\
&+ (A_{(n-3)k} - A_{(n-2)k}) \ln(L_{n-2}) + ..... \\
&+ (A_{(n-2)k} - A_{(n-1)k} - A_{nk}) \ln(L_{n-1}) + (A_{nk} + A_{(n-1)k}) \ln(L_{nk})
\end{aligned}$$

Writing the above equation in a matrix form we get

$$\begin{pmatrix} \phi_2 \\ \phi_3 \\ \vdots \\ \phi_k \\ \vdots \\ \phi_{n-2} \\ \phi_{n-1} \end{pmatrix} = \begin{pmatrix} -A_{02}-A_{12} & \cdots & A_{n2}+A_{(n-1)2} \\ -A_{03}-A_{13} & \cdots & A_{n3}+A_{(n-1)3} \\ \vdots & \ddots & \vdots \\ -A_{0k}-A_{1k} & \cdots & A_{nk}+A_{(n-1)k} \\ \vdots & \ddots & \vdots \\ A_{n,(n-2)}-A_{1,(n-2)} & \cdots & A_{n,(n-2)}+A_{(n-1),(n-2)} \\ A_{n,(n-1)}-A_{1,(n-1)} & \cdots & A_{n,(n-1)}+A_{(n-1),(n-1)} \end{pmatrix} \begin{pmatrix} \ln(L_1) \\ \ln(L_2) \\ \vdots \\ \ln(L_{k-1}) \\ \vdots \\ \ln(L_{n-1}) \\ \ln(L_n) \end{pmatrix}$$

Its can be observed that the above matrix is of the form

$$\phi_k = a_{ki} M_i \quad (9.8)$$

## 9.2 Appendix B

### Derivation: Cubic Constrains

For simplicity initially we do the derivation only for equation  $u_i(\omega)$ , and the result is extended to  $v_i(\omega)$ .

So  $u_i(\omega)$  is given by

$$u_i(\omega) = p_{i1}\omega^3 + p_{i2}\omega^2 + p_{i3}\omega + p_{i4} \quad (9.9)$$

Since  $u_i(\omega)$  is a third order polynomial, it is constructed over four coordinates pairs denoted by  $(\omega_{i0}, M_{i0})$ ,  $(\omega_{i1}, M_{i1})$ ,  $(\omega_{i2}, M_{i2})$  and  $(\omega_{i3}, M_{i3})$ .

Next task is to generate successively higher degree polynomial at a specific point, in this case  $i$  for which  $\omega \in [\omega_i, \omega_{i+1}]$ . This is done using the method of *Newton's Divided*

differences [7, 47].

Suppose that  $P_n(x)$  is the Lagrange polynomial that agrees with the function  $f$  at the distinct numbers  $x_0, x_1, \dots, x_n$ . The divided differences of  $f$  with respect to  $x_0, x_1, \dots, x_n$  are used to express  $P_n(x)$  in the form

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots \quad (9.10)$$

$$+ a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}), \quad (9.11)$$

for appropriate constants  $a_0, a_1, \dots, a_n$ .

To determine the first of these constants  $a_0$ , note that if  $P_n(x)$  is written in the form of equation's 7.10 and 7.11 at  $x_0$  leaves only the constant term  $a_0$ ; that is,

$$a_0 = P_n(x) = f(x_0).$$

Similarly, when  $P(x)$  is evaluated at  $x_1$ , the only nonzero terms in the evaluation of  $P_n(x_1)$  are the constant and linear terms,

$$f(x_0) + a_1(x_1 - x_0) = P_n(x_1) = f(x_1);$$

so

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

We now introduce the divided-difference notation, which is related to Aitken's  $\Delta^2$  notation, this is used to accelerate the convergence of the sequence. The zeroth divided difference of the function  $f$  with respect to  $x_i$ , denoted  $f[x_i]$ , is simply the value of  $f$  at  $x_i$ ,

$$f[x_i] = f(x_i).$$

The remaining divided differences are defined inductively, the first divided difference of  $f$  with respect to  $x_i$  and  $x_{i+1}$  is denoted as  $f[x_i, x_{i+1}]$  and is defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}.$$

The second divided difference,  $f[x_i, x_{i+1}, x_{i+2}]$ , is defined as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Similarly, after the  $(k - 1)$ st divided differences,

$$f[x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k-1}] \quad \text{and} \quad f[x_{i+1}, x_{i+2}, \dots, x_{i+k-1}, x_{i+k}],$$



have been determined, the  $k$ th divided difference relative to

$x_i, x_{i+1}, x_{i+2}, \dots, x_{i+k}$  is given by

$$f[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

With this result equation( $a_1$ ) can be rewritten as  $a_1 = f[x_0, x_1]$ , and the interpolation polynomial in equation ( $P_n(x)$ ) is written as

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + a_2(x - x_0)(x - x_1) \\ + \dots + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

As a result of the evaluation of  $a_0$  and  $a_1$ , the required constants are

$$a_k = f[x_0, x_1, x_2, \dots, x_k],$$

for each  $k = 0, 1, \dots, n$ . So  $P_n(x)$  can be rewritten as[hild]

$$P_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, x_1, \dots, x_k](x - x_0) \dots (x - x_{k-1}).$$

The value of  $f[x_0, x_1, \dots, x_k]$  is independent of the order of the numbers  $x_0, x_1, \dots, x_k$ . This equation is known as *Newton's interpolatory divided difference formula*. This can also be written as

$$P(x) = \sum_{i=0}^n F_{i,i} \prod_{j=0}^{i-1} (x - x_j).$$

This result can be written in a tabular form showing the first, second and the third divided difference of the equation.

$x f(x)$	First	Second	Third
$x_0 f[x_0]$			
$x_1 f[x_1]$	$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$	$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$
$x_2 f[x_2]$	$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$	$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$
$x_3 f[x_3]$	$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$	$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	$f[x_2, x_3, x_4, x_5] = \frac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2}$
$x_4 f[x_4]$	$f[x_3, x_4] = \frac{f[x_4] - f[x_3]}{x_4 - x_3}$	$f[x_3, x_4, x_5] = \frac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3}$	
$x_5 f[x_5]$	$f[x_4, x_5] = \frac{f[x_5] - f[x_4]}{x_5 - x_4}$		

Now using this result, equation ( $u_i(\omega)$ ) can be rewritten as

$$u_i(\omega) = f[\omega_{i0}] + f[\omega_{i0}, \omega_{i1}](\omega - \omega_{i0}) + f[\omega_{i0}, \omega_{i1}, \omega_{i2}](\omega - \omega_{i0})(\omega - \omega_{i1}) \\ + f[\omega_{i0}, \omega_{i1}, \omega_{i2}, \omega_{i3}](\omega - \omega_{i0})(\omega - \omega_{i1})(\omega - \omega_{i2})$$

where

$$\begin{aligned}
f[\omega_{i0}] &= M_{i0} \\
f[\omega_{i0}, \omega_{i1}] &= \frac{f[\omega_{i1}] - f[\omega_{i0}]}{\omega_{i1} - \omega_{i0}} \\
&= A_{i0}M_{i0} - A_{i1}M_{i1} \\
f[\omega_{i0}, \omega_{i1}, \omega_{i2}] &= \frac{f[\omega_{i1}, \omega_{i2}] - f[\omega_{i0}, \omega_{i1}]}{\omega_{i2} - \omega_{i0}} \\
&= B_{i0}M_{i0} + B_{i1}M_{i1} + B_{i2}M_{i2} \\
f[\omega_{i0}, \omega_{i1}, \omega_{i2}, \omega_{i3}] &= \frac{f[\omega_{i1}, \omega_{i2}, \omega_{i3}] - f[\omega_{i0}, \omega_{i1}, \omega_{i2}]}{\omega_{i3} - \omega_{i0}} \\
&= C_{i0}M_{i0} + C_{i1}M_{i1} + C_{i2}M_{i2} + C_{i3}M_{i3}
\end{aligned}$$

and

$$\begin{aligned}
A_{i0} &= \frac{1}{\omega_{i0} - \omega_{i1}} & A_{i1} &= \frac{1}{\omega_{i1} - \omega_{i0}} \\
B_{i0} &= \frac{1}{(\omega_{i1} - \omega_{i0})(\omega_{i2} - \omega_{i0})} & B_{i1} &= \frac{1}{(\omega_{i2} - \omega_{i1})(\omega_{i1} - \omega_{i0})} \\
B_{i2} &= \frac{1}{(\omega_{i2} - \omega_{i1})(\omega_{i2} - \omega_{i0})} \\
C_{i0} &= \frac{1}{(\omega_{i0} - \omega_{i1})(\omega_{i2} - \omega_{i0})(\omega_{i3} - \omega_{i0})} \\
C_{i1} &= \frac{(\omega_{i3} - \omega_{i1})^{-1}}{(\omega_{i3} - \omega_{i0})(\omega_{i2} - \omega_{i1})} + \frac{(\omega_{i1} - \omega_{i0})^{-1}}{(\omega_{i2} - \omega_{i0})(\omega_{i3} - \omega_{i0})} + \frac{(\omega_{i2} - \omega_{i1})^{-1}}{(\omega_{i2} - \omega_{i0})(\omega_{i3} - \omega_{i0})} \\
C_{i2} &= \frac{(\omega_{i2} - \omega_{i3})^{-1}}{(\omega_{i3} - \omega_{i1})(\omega_{i3} - \omega_{i0})} + \frac{(\omega_{i1} - \omega_{i3})^{-1}}{(\omega_{i3} - \omega_{i0})(\omega_{i2} - \omega_{i1})} + \frac{(\omega_{i1} - \omega_{i2})^{-1}}{(\omega_{i2} - \omega_{i0})(\omega_{i3} - \omega_{i0})} \\
C_{i3} &= \frac{1}{(\omega_{i3} - \omega_{i2})(\omega_{i3} - \omega_{i1})(\omega_{i3} - \omega_{i0})}
\end{aligned}$$

Now define

$$u_i(\omega) = u_{i0}(\omega) + u_{i1}(\omega) + u_{i2}(\omega) + u_{i3}(\omega)$$

where

$$\begin{aligned}
u_{i0}(\omega) &= f[\omega_{i0}] \\
u_{i1}(\omega) &= f[\omega_{i0}, \omega_{i1}](\omega - \omega_{i0}) \\
u_{i2}(\omega) &= f[\omega_{i0}, \omega_{i1}, \omega_{i2}](\omega - \omega_{i0})(\omega - \omega_{i1}) \\
u_{i3}(\omega) &= f[\omega_{i0}, \omega_{i1}, \omega_{i2}, \omega_{i3}](\omega - \omega_{i0})(\omega - \omega_{i1})(\omega - \omega_{i2})
\end{aligned}$$

Based on the above results, by substituting the equations we get

$$u_{i0}(\omega) = M_{i0} = K_{i0}M_{i0}$$

$$u_{i1}(\omega) = (k_{i1}M_{i0} + k_{i2}M_{i1})(k_{i3}M_{i0} + k_{i4}M_{i1})\omega$$

$$u_{i2}(\omega) = (k_{i5}M_{i0} + k_{i6}M_{i1} + k_{i7}M_{i2})$$

$$+ \omega(k_{i8}M_{i0} + k_{i9}M_{i1} + k_{i10}M_{i2})$$

$$+ \omega^2(k_{i11}M_{i1} + k_{i28}M_{i2} + k_{i29}M_{i3})$$

$$u_{i3}(\omega) = (k_{i14}M_{i0} + k_{i15}M_{i1} + k_{i16}M_{i2} + k_{i17}M_{i3})$$

$$+ \omega(k_{i18}M_{i0} + k_{i19}M_{i1} + k_{i20}M_{i2} + k_{i21}M_{i3})$$

$$+ \omega^2(k_{i22}M_{i0} + k_{i23}M_{i1} + k_{i24}M_{i2} + k_{i25}M_{i3})$$

$$+ \omega^3(k_{i26}M_{i0} + k_{i27}M_{i1} + k_{i28}M_{i2} + k_{i29}M_{i3})$$

where

$$k_{i0} = 1, k_{i1} = (-\omega_{i0})A_{i0}, k_{i2} = (-\omega_{i0})A_{i1}, k_{i3} = A_{i0}, k_{i4} = A_{i1},$$

$$k_{i5} = (\omega_{i0}\omega_{i1})B_{i0}, k_{i6} = (\omega_{i0}\omega_{i1})B_{i1}, k_{i7} = (\omega_{i0}\omega_{i1})B_{i2}, k_{i8} = -(\omega_{i0} + \omega_{i1})B_{i0},$$

$$k_{i9} = -(\omega_{i0} + \omega_{i1})B_{i1}, k_{i10} = -(\omega_{i0} + \omega_{i1})B_{i2}, k_{i11} = B_{i0}, k_{i12} = B_{i1}, k_{i13} = B_{i2},$$

$$k_{i14} = (-\omega_{i0}\omega_{i1}\omega_{i2})C_{i0}, k_{i15} = (-\omega_{i0}\omega_{i1}\omega_{i2})C_{i1}, k_{i16} = (-\omega_{i0}\omega_{i1}\omega_{i2})C_{i2},$$

$$k_{i17} = (-\omega_{i0}\omega_{i1}\omega_{i2})C_{i3}, k_{i18} = (\omega_{i1}\omega_{i2} + \omega_{i0}\omega_{i2} + \omega_{i0}\omega_{i1})C_{i0},$$

$$k_{i19} = (\omega_{i1}\omega_{i2} + \omega_{i0}\omega_{i2} + \omega_{i0}\omega_{i1})C_{i1}, k_{i20} = (\omega_{i1}\omega_{i2} + \omega_{i0}\omega_{i2} + \omega_{i0}\omega_{i1})C_{i2},$$

$$k_{i21} = (\omega_{i1}\omega_{i2} + \omega_{i0}\omega_{i2} + \omega_{i0}\omega_{i1})C_{i3}, k_{i22} = -(\omega_{i0} + \omega_{i1} + \omega_{i2})C_{i0},$$

$$k_{i23} = -(\omega_{i0} + \omega_{i1} + \omega_{i2})C_{i1}, k_{i24} = -(\omega_{i0} + \omega_{i1} + \omega_{i2})C_{i2},$$

$$k_{i25} = -(\omega_{i0} + \omega_{i1} + \omega_{i2})C_{i3}, k_{i26} = C_{i0}, k_{i27} = C_{i1}, k_{i28} = C_{i2}, k_{i29} = C_{i3},$$

Based on the original cubic polynomial equation of  $u_i(\omega)$  in terms of  $p_{ix}$ , we get the

coefficients as

$$\begin{aligned}
p_{i1} &= (k_{i26})M_{i0} + (k_{i27})M_{i1} + (k_{i28})M_{i2} + (k_{i29})M_{i3} \\
&= a_{i0}M_{i0} + a_{i1}M_{i1} + a_{i2}M_{i2} + a_{i3}M_{i3} \\
p_{i2} &= (k_{i11} + k_{i22})M_{i0} + (k_{i12} + k_{i23})M_{i1} + (k_{i13} + k_{i24})M_{i2} + (k_{i25})M_{i3} \\
&= b_{i0}M_{i0} + b_{i1}M_{i1} + b_{i2}M_{i2} + b_{i3}M_{i3} \\
p_{i3} &= (k_{i3} + k_{i8} + k_{i18})M_{i0} + (k_{i4} + k_{i9} + k_{i19})M_{i1} + (k_{i10} + k_{i20})M_{i2} \\
&\quad + (k_{i21})M_{i3} \\
&= c_{i0}M_{i0} + c_{i1}M_{i1} + c_{i2}M_{i2} + c_{i3}M_{i3} \\
p_{i4} &= (k_{i0} + k_{i1} + k_{i5} + k_{i14})M_{i0} + (k_{i2} + k_{i6} + k_{i15})M_{i1} + (k_{i7} + k_{i16})M_{i2} \\
&\quad + (k_{i17})M_{i3} \\
&= d_{i0}M_{i0} + d_{i1}M_{i1} + d_{i2}M_{i2} + d_{i3}M_{i3}
\end{aligned}$$

Now the polynomial can be written as a matrix relation in terms of  $p$ ,  $M$  and a constant matrix  $A_i$  as

$$[p_{i1} \ p_{i2} \ p_{i3} \ p_{i4}]^T = A_i [M_{i-1} \ M_i \ M_{i+1} \ M_{i+2}]^T \quad (9.12)$$

where

$$A_i = \begin{pmatrix} a_{i0} & a_{i1} & a_{i2} & a_{i3} \\ b_{i0} & b_{i1} & b_{i2} & b_{i3} \\ c_{i0} & c_{i1} & c_{i2} & c_{i3} \\ d_{i0} & d_{i1} & d_{i2} & d_{i3} \end{pmatrix}$$

Now writing the cubic polynomial for  $v_i(\omega)$  in the same form we get

$$[q_{i1} \ q_{i2} \ q_{i3} \ q_{i4}]^T = A_i [\phi_{i-1} \ \phi_i \ \phi_{i+1} \ \phi_{i+2}]^T \quad (9.13)$$

## 9.3 Appendix C

### 9.3.1 M-file to demonstrate controller design using QFT method.

```

% Define the range of frequencies of interest & of general
frequency range
m_circ_range=[0.0001 0.01 0.1 0.5 0.7 1 1.4 2.8];
w_interest_track=[.5 1 2 3 5 10 40 80]; % for illustration
purposes
m_circ_value=1.2; % m_circle for UHFB

```

```

% For disturbance rejection low frequency range needed
up_limit_w_dis=80; % upper limit of freq. for disturbance
rejection
trans1=w_interest_track <= up_limit_w_dis;
index_dis=find(trans1);
w_interest_dis=w_interest_track(index_dis); %dist. rej.
freq.of interest
w_range=logspace(-2,2.5,100); % frequency range
nw_mark=size(w_interest_track,2); % size of frequency
range

% Define plant uncertainty
k=[1:1:10];
a=[1:1:10];
num_p=[0 0 1]; % nominal plant's numerator
den_p=[1 1 0]; % nominal plant's denominator

% Define input specifications:
num_u = [0.582 11.641];

den_u = [1 2.66 11.641];
num_l = 2.75*20;

den_l = [1 22.65 55.75 55];

% Derive the frequency response of the upper and lower
models
% ...for the general range of frequencies
[mu,pu]=bode(num_u,den_u,w_range);
mu=20*log10(mu); % translation in dB (*) upper bound
[m1,pl]=bode(num_l,den_l,w_range);
m1=20*log10(m1); % (*)lower bound
% ...for the frequencies of interest
[mu_m,pu_m]=bode(num_u,den_u,w_interest_track);
mu_m=20*log10(mu_m); % (*)upper bound
[m1_m,pl_m]=bode(num_l,den_l,w_interest_track);
m1_m=20*log10(m1_m); % (*) lower bound
spec=mu_m-m1_m; % tracking specs
(difference upper_bound_mark-lower_bound_mark)
% 'mark' indicates point at the frequencies of interest
%
% Derive the uncertainty templates (use convex hull to
reduce the computational
% burden of the algorithm)
c_initial=temp_cs1(k',a',w_interest_track');
% uncertainty templates
c_convex=c_hull1(c_initial); % convex hull
% Nominal open loop
nom_ol=temp_cs1(1,1,w_range');
nom_ol_mark=temp_cs1(1,1,w_interest_track');

% -----

```

```

% Introducing 2nd order disturbance rejection
%
do=1; % step input disturbance
ap=.1; % maximum allowed gain that dist. can reach after
ts
tx=2; % settling time for the dist. input in sec
pu=25; % max. percent overshoot of the dist. input
[a,b,num_model,den_model] = dop_mod2(do,ap,tx,pu);
% -----
% The disturbance rejection is translated in the frequency
domain
[mag_dist,phase_dist]=bode(num_model,den_model,w_range);
% storing the coordinates
% %of the bode plot
% storing the coordinates of the mark frequencies of the
bode plot
[mag_dist_mark,phase_dist_mark]=bode(num_model,den_model,
w_interest_dis);
%
% Finding the maximum rejection bound (db)
bound_max=(20*log10(mag_dist_mark));
% -----
b_dis=[];
for i=1:size(w_interest_dis,2)
    disp(['calculating dis. temp. for frequency ',
int2str(i)]);
    [bd,phid]=dis_bnds(c_convex(i,:),c_convex(i,1),-80,80,
-360,0,91,bound_max(i),.1);
    b_dis=[b_dis;bd];
end
%
% -----
% Find Horowitz templates and u-contour
b_hor=[]; % Horowitz templates creation
for i=1:size(w_interest_track,2)
    disp(['Calculating Horowitz templates for frequency ',
int2str(i)]);
    [b,phi]=hr_bnds1(c_convex(i,:),c_convex(i,1),-80,80,
-360,0,91,spec(i),.1);
    b_hor=[b_hor;b];
end
%
% HFC (High Frequency Contour) M=m_circ_value and m-circle
m_circ_value
nphi=size(phi,2); % size of number of phases
u_templ=zeros(nw_mark,nphi); % initialise template for
m-circle
for i=1:nw_mark
    ut=m_cir(m_circ_value,phi,-80); % m-circle m_circ_value
    ut=ut(1,:);
    u_templ(i,:)=ut;
end

```

```

hfc=hf_bound(20*log10(m_circ_value),40,0,0); % hfc

% Find overall bounds, nominal open loop (for general range of
frequencies and for
% marked frequencies)
b_all1=zeros(size(b_dis)); % Merged templates between dis
and hor. templ.
b_all=zeros(size(b_hor)); % Overall templates
for i=1:size(b_hor,1)
    if i <= size(b_dis,1)
        b_all1(i,:)=max(b_hor(i,:),b_dis(i,:));
        b_all(i,:)=max(b_all1(i,:),u_templ(i,:));
    else
        b_all(i,:)=max(b_hor(i,:),u_templ(i,:));
    end
end
m_grid1(-270,0,40,[0.001 0.01 0.1 0.5 0.7 1 1.25 1.4 2 4])
title('Uncertainty templates for the range of chosen
frequencies')
plot(c_convex.','k')
plot(c_convex.','k+')
plot(nom_ol_mark,'ko'); % nominal plant for marked
frequencies
nom_ol_mark_text=nom_ol_mark+3;
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
% Plot the freq. response of the dist. rej model figure 2
figure % create figure for the disturbance frequency
response
semilogx(w_range,20*log10(mag_dist),'k-',
w_interest_dis,20*log10
(mag_dist_mark),'ko');
grid;
coordy=[20*log10(mag_dist_mark)+3];
coordx=w_interest_dis;
for i=1:size(w_interest_dis,2)
    text(coordx(i),coordy(i),num2str(w_interest_dis(i)))
end
title ('2nd order disturbance rejection model frequency
reponse');
xlabel('ang. freq. rads/s');
ylabel('magnitude dB');
% Nichols chart including all details 3
m_grid1(-360,0,40,[0.001 0.01 0.1 0.5 0.7 1 1.25 1.4 2 4])
title('Detailed representation of all parameters')
plot(nom_ol,'k--');
plot(nom_ol_mark,'go');
plot(phi,b_hor.','k-');

```

```

plot(phid,b_dis.','b-');
plot(hfc,'r-')
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
% Nichols chart including overall bounds,hfc,m_circle
figure 4
m_grid1(-360,0,40,[0.001 0.01 0.1 0.5 0.7 1 1.25 1.4 2 4])
title('Overall bounds and ol system before and after
controller design')

plot(nom_ol,'k--');
plot(nom_ol_mark,'go');
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end

w=logspace(-0.6,3,100);

wm=w_interest_track;
plot(phi,b_all.','b-');
plot(hfc,'r-')
nump=[0 0 1];
denp=[1 1 0];

%num=[1.212 216.9];

%den=[1 36.76 35.7 0];

%num=[1 159.6150];

%den=[1 41.1 40.1 0];

%num=25*[1 74.86];
num=[1 74.86];
den=[1 86.49 85.49 0];

[mag,phase]=bode(num,den,w); % freq. response
[magm,phasem]=bode(num,den,wm); % freq. response for marked
frequencies
mag=20*log10(mag)+20*log10(w'); % translation in dBs (*)
magm=20*log10(magm)+20*log10(wm'); % (*)
phase=phase+90; % introducing +90 due to the existence of
1/s
phasem=phasem+90;
ol=phase+j*mag; % open loop
olmark=phasem+j*magm;

```



```

    %m_grid1(-360,0,40,[0.001 0.01 0.1 0.5 0.7 1 1.25 1.4 2 4])
    plot(ol,'r-')
    plot(olmark,'go')

%-----%
% Step 3: CL with Pre-filter (2nd order)
%-----%
% Plot specifications, figure 5
figure
semilogx(w_range,mu,'k-',w_range,ml,'k-',w_interest_track,
mu_m,'ko',
w_interest_track,ml_m,'ko')
grid,hold
% Settings:
A=1;
k_r=[1 20 50 100];
a_r=[1 20 50 100];
wc1=3.5; % 1st cut-off
wc2=7.0; % 2nd cut-off
% Frequency response at all frequencies
[magsys,phasesys]=fr_cl_pf(A,wc1,wc2,k_r,a_r,
kd_opt,ki_opt,
kp_opt,w_range);
% Frequency response at marked frequencies
[magsys_m,phasesys_m]=fr_cl_pf(A,wc1,wc2,k_r,
a_r,kd_opt,
ki_opt,kp_opt,
w_interest_track);
% Add the new information in figure 5
semilogx(w_range,magsys,'k--',w_interest_track,
magsys_m,'ko')
title('Frequency response of cl designed system')
xlabel('ang. frequency rads/s');
ylabel('magnitude dB');
% Frequency response of Tyd at all frequencies
[magsysd,phasesysd]=dis_r_fr(k_r,a_r,kd_opt,ki_opt,
kp_opt,w_range);
% Frequency response of Tyd at marked frequencies
[magsys_dm,phasesys_dm]=dis_r_fr(k_r,a_r,kd_opt,
ki_opt,kp_opt,
w_interest_dis);
% Add the new information in figure 6 with the
disturbance spec.
figure
semilogx(w_range,20*log10(mag_dist),'k-',w_interest_dis,
20*log10(mag_dist_mark),'ko');
grid;
title ('Rejection of the disturbance in the
frequency domain');
xlabel('ang. freq. rads/s');
ylabel('magnitude dB');hold;

```

```

semilogx(w_range,magsysd,'k--',w_interest_dis,magsys_dm,'
ko')
%
% Find the differences of the bounds at the given freqs
difference_tracking=zeros(size(magsys_m,1),1);
dis_gain=zeros(size(magsys_dm,1),1);
for i=1:size(magsys_m,1)
    max_limit=max(magsys_m(i,:));
    min_limit=min(magsys_m(i,:));
    difference_tracking(i)=max_limit-min_limit;
end
for i=1:size(magsys_dm,1)
    max_limit=max(magsys_dm(i,:));
    dis_gain(i)=max_limit;
end
% Now the step responses...
t=[0:1000]/200; % simulation time (5 secs)
y_upper=step(num_u,den_u,t); % specifications
y_lower=step(num_l,den_l,t); % specifications
% System designed
[stepall,tsim]=step_sim(k_r,a_r,kd_opt,ki_opt,kp_opt,1,
wc1,wc2,t);
% Plot responses
figure
grid,hold
title('Step responses of designed cl system and
specifications');
xlabel('Simulation time secs');
ylabel('Amplitude');
plot(t,y_upper,'k-',t,y_lower,'k-',t,stepall,'k--');
% Disturbance rejection in the time domain
% Derived model..
d_rej_model=step(num_model,den_model,t);
% Designed system
dist_resp=dist_rej(k_r,a_r,kd_opt,ki_opt,kp_opt,t);
% Plot result
figure
grid,hold
title('Disturbance rejection in time domain');
xlabel('Simulation time secs');
ylabel('Amplitude');
plot(t,d_rej_model,'k-.',t,dist_resp,'k-');

```

### 9.3.2 M-file to demonstrate Bode approximation.

```

n=size(omega,2); % no of frequencies (rows)
%
if n ~= size(gains,2) | n <= 4
    disp('error in bode_approx.m ...');
    phase=[];
    weights=[];

```

```

        return
    end
%
% Initialise matrices:
a=zeros(n-1,n-1);
b=zeros(1,n-1);
c=zeros(1,n-1);
%
for k=2:n-1
b(k)=quad8('log_coth',log(omega(n)/omega(k)),20);
b(k)=b(k)/(pi*log(omega(n)/omega(n-1)));
c(k)=quad8('log_coth',-20,log(omega(1)/omega(k)));
c(k)=c(k)/(pi*log(omega(2)/omega(1)));
for i=1:n-1
    if i==k
        temp=quad8('log_coth',log(omega(i+1)/omega(k)),20);
        a(i,k)=pi*pi/4-temp;
    elseif i+1 == k
        temp=quad8('log_coth',-log(omega(i)/omega(k)),20);
        a(i,k)=pi*pi/4-temp;
    elseif omega(i) < omega(k) & omega(i+1) > omega(k)
        lim1=-log(omega(i)/omega(k));
        lim2=log(omega(i+1)/omega(k));
        temp=quad8('log_coth',min(lim1,lim2),max(lim1,lim2));
        a(i,k)=pi*pi/2-temp;
    else
        a(i,k)=quad8('log_coth',log(omega(i)/omega(k)),
            log(omega(i+1)/omega(k)));
        end % if
        a(i,k)=a(i,k)/(pi*log(omega(i+1)/omega(i)));
    end % for
end % for
%
% Assemble weights and reconstruct phase
weights=zeros(n-2,n);
weights(:,1)=(c(2:n-1))'-(a(1,2:n-1))';
weights(:,2)=-(c(2:n-1))'+(a(1,2:n-1))'-(a(2,2:n-1))';
weights(:,n-1)=-(b(2:n-1))'+(a(n-2,2:n-1))'-(
    a(n-1,2:n-1))';
weights(:,n)=(b(2:n-1))'+(a(n-1,2:n-1))';
%
for k=3:n-2
    weights(:,k)=(-a(k,2:n-1))'+(a(k-1,2:n-1))';
end
%
weights=weights*((180*log(10))/(20*pi));
phase=weights*gains';
phase=phase';

```

### 9.3.3 The function used to generate the uncertainty templates

```
function c=temp_cs1(k,a,w)
% Creates the uncertain template of a fixed system  $G(s)=k*a/s*(s+a)$ 
% Inputs: k,a: uncertain parameters
%          w: array of frequencies of interest
% Output:  c: uncertainty templates (matrix form)
%
% Find the sizes of the uncertainty parameters
nk=size(k,1);
na=size(a,1);
nw=size(w,1); % size of frequencies of interest
% Create (initialise) uncertainty template
c=zeros(nw,nk*na);
for i=1:nk
    for j=1:na
        num=k(i)*a(j);
        den=[1 a(j) 0];
        [mag,phase]=bode(num,den,w); % frequency response of system
        mag=20*log10(mag); % dB translation of magnitude
        temp=phase+sqrt(-1)*mag; % temporary stores each template
        c(:,j+(i-1)*na)=temp; % fill each column of the template
    end
end
%
%-----end of temp_cs1.m-----
```

### 9.3.4 M-file to generate the convex hull with the sub function

```
function array_out=c_hull1(array_in)
%
% Function computes the convex hull of complex data in row array_in.
% The (consecutive) vertices of the complex hull appear in array_out.
% The first and last points of the hull are identical.
% If array_in is a matrix the convex hulls of all rows are computed.
% The results appear in array_out with (possibly) repeated end points
% for dimension compatibility.
%
% Input : array_in: input array of which convex hull is derived
% Output: array_out: convex hull of array_in
%
disp('Calculating Convex Hull ...');
n_rows=size(array_in)*[1;0];
n_cols=size(array_in)*[0;1];
nmax=0;
%
for i_r=1:n_rows
    disp(['Row ',int2str(i_r),' ...']);
    re_sum=real(array_in(i_r,:));
    im_sum=imag(array_in(i_r,:));
%
% Remove identical points from re_sum,im_sum; save in re_sum1, im_sum1
%
```

```

re_sum1=zeros(1,size(re_sum)*[0;1]);
im_sum1=zeros(1,size(im_sum)*[0;1]);
re_sum1(1,1)=re_sum(1,1);
im_sum1(1,1)=im_sum(1,1);
index=0;
%
for i=2:max(size(re_sum))
    log1=any(re_sum1+sqrt(-1).*im_sum1==re_sum(i)+sqrt(-1).*im_sum(i));
    log1=any(re_sum1==re_sum(i)).*any(im_sum1==im_sum(i));
    if all(log1==0)==1
        re_sum1(1,i-index)=re_sum(1,i);
        im_sum1(1,i-index)=im_sum(1,i);
    else
        index=index+1;
    end
end
%
re_sum1=re_sum1(:,1:max(size(re_sum))-index);
im_sum1=im_sum1(:,1:max(size(im_sum))-index);
%
% Determine point inside hull (centroid of 3 points)
%
[x1,index1]=min(re_sum1);y1=im_sum1(index1);
[x2,index2]=max(re_sum1);y2=im_sum1(index2);
[y3,index3]=min(im_sum1);x3=re_sum1(index3);
%
jjj=sqrt(-1);
inter_point=((x1+x2+x3)/3)+jjj*((y1+y2+y3)/3);
%
% Calculate phase of input array; (between 0 and 2*pi)
%
ph=zeros(1,max(size(re_sum1)));
for i=1:max(size(re_sum1))
    ph(i)=angle(re_sum1(i)+jjj*im_sum1(i)-inter_point);
    if im_sum1(i)-imag(inter_point) < 0
        ph(i)=ph(i)+2*pi;
    end
end
%
[ph_s,index]=sort(ph);
re_sum2=re_sum1(index);
im_sum2=im_sum1(index);
%
% Find point of minimum y-coordinate and re-order sequence in
% counter-clockwise order with it as starting point.
%
[y_min,index]=min(im_sum2);
if index > 1
    re_s=[re_sum2(index:max(size(re_sum2))) re_sum2(1:index-1)];
    im_s=[im_sum2(index:max(size(re_sum2))) im_sum2(1:index-1)];
end
%

```

```

    re1_s=[re_s re_s(1)];
    im1_s=[im_s im_s(1)];
    c1_s=re1_s+jjj.*im1_s;
%
    v=c1_s(1);
    ind_v=1;
    while next(c1_s,ind_v) ~= c1_s(1)
        if l_turn1(v,next(c1_s,ind_v),next(c1_s,ind_v+1)) == 1
            v=next(c1_s,ind_v);
            ind_v=ind_v+1;
        else
            c1_s(ind_v+1)=[];
            if v ~= c1_s(1)
                v=prev(c1_s,ind_v);
                ind_v=ind_v-1;
            end
        end
    end
    end
%
    eval(['array_out',int2str(i_r),'=c1_s;']);
    if max(size(c1_s)) > nmax
        nmax=max(size(c1_s));
    end
end
%
array_out=zeros(n_rows,nmax);
for i=1:n_rows
    eval(['n_temp=max(size(array_out',int2str(i),'));']);
    eval(['last_el=array_out',int2str(i),'(n_temp);']);
    eval(['array_out',int2str(i),...
        '=[array_out',int2str(i),' last_el.*ones(1,nmax-n_temp)];']);
    eval(['array_out(',int2str(i),',:)=array_out',int2str(i),',:');']);
end
%
%----- end of c_hull1.m -----
%
function out=l_turn1(point1,point2,point3)
%
% Produces 1 if point 3 lies to the left of directed line
% point1 to point2 and 0 otherwise. Points must be complex
% numbers.
%
p21=point2-point1;
p31=point3-point1;
alpha=real(p21);beta=imag(p21);
theta=atan2(-beta,alpha);
%
if cos(theta)*alpha-sin(theta)*beta < 0
    theta=-theta;
end
%
u=[cos(theta) -sin(theta);sin(theta) cos(theta)];

```

```

    p21_t=u*[real(p21);imag(p21)];
    p21_t=p21_t(1)+sqrt(-1)*p21_t(2);
    p31_t=u*[real(p31);imag(p31)];
    p31_t=p31_t(1)+sqrt(-1)*p31_t(2);
%
    if angle(p31_t) > 0
        out=1;
    else
        out=0;
    end
%
%----- end of l_turn1.m -----
%
    function point_o=next(array,point_i)
% Function gives next point in array; If point is last, empty string
% is returned
%
    if point_i < max(size(array))
        point_o=array(point_i+1);
    else
        point_o=[];
    end
%
%----- end of next.m -----
    function point_o=prev(array,point_i);
% Function gives previous point in array; If point is first, empty string
% is returned
%
    if point_i > 1
        point_o=array(point_i-1);
    else
        point_o=[];
    end
%
%----- end of prev.m -----

```

### 9.3.5 M.file to calculate the Horowitz templates

Note that this relies on an additional function (chk-span.m) which calculates whether an uncertainty template, translated to a certain position on the Nichols chart, satisfies the closed-loop robust tracking specifications.

```

function [b,ph_out]=hr_bnds1(fr_p,ref_p,min_m,max_m,min_p,max_p,...
ph_ps,bound,tol_m)
%
% Function determines the Horowitz regions (minimum gain regions
% in Nichols chart at which template variations lie within
% specified M-circle tolerance). Function uses bisection algorithm
% for given phase span and for specified magnitude tolerance.
%
% fr_p : Uncertainty template (row vector of complex numbers)

```

```

%      on Nichols chart. Real part describes the phase in
%      degrees, imaginary part is equal to 20*log10(magnitude)
% ref_p : Reference point on which translation is based (could
%      be arbitrary point inside template, first point in
%      template etc.)
% min_m : Minimum magnitude open-loop level in db
% max_m : Maximum magnitude open-loop level in db
% min_p : Minimum open-loop phase in degrees (typically -180)
% max_p : Maximum open-loop phase in degrees (typically 0)
% ph_ps : Number of phase points linearly spaced between min
%      and max values (>=2)
% bound : Minimum gain span (db) that closed-loop must satisfy
%      for all members of uncertainty template
% tol_m : Magnitude contour tolerance (db)
% b      : Horowitz contour (db's)
% ph_out: (Linearly-spaced) phase vector corresponding to b (degrees)
%-----
% Error messages:
%
% if size(fr_p)*[1;0] ~= 1
%     error('First variable should be row vector ...');
% end
% if min_m >= max_m
%     error('Minimum magnitude level should not exceed maximum level ...');
% end
% if min_p >= max_p
%     error('Minimum phase level should not exceed maximum level ...');
% end
% if tol_m <= 0
%     error('Magnitude tolerance variable should be positive ...');
% end
%
% step_p=(max_p-min_p)/(ph_ps-1); % phase step
% mag_ref=imag(ref_p); % magnitude of reference point in db's
% ph_ref=real(ref_p); % phase of reference points in db's
% b=zeros(1,ph_ps); % initialize Horowitz contour
% n_tmpl=size(fr_p)*[0;1]; % number of points in uncertainty templates
% max_iter=100; % maximum number of iterations
%
% for i=1:ph_ps
%
%     disp(['Phase point # ',int2str(i)]);
%     phase=min_p+(i-1)*step_p;
%     templ_low=fr_p+(phase-ph_ref+sqrt(-1)*(min_m-mag_ref)).*...
%     ones(1,n_tmpl);
%     templ_high=fr_p+(phase-ph_ref+sqrt(-1)*(max_m-mag_ref)).*...
%     ones(1,n_tmpl);
%
%     if chk_span(templ_low,bound,0,0) == 1
%         b(i)=min_m;
%     elseif chk_span(templ_high,bound,0,0) == 0
%         b(i)=max_m;

```



```

        else
            delta=max_m-min_m;
            m_high=max_m;
            m_low=min_m;
            n_iter=1;
            while delta > tol_m & n_iter <= max_iter
                n_iter=n_iter+1;
                m_cand=0.5*(m_high+m_low);
                if chk_span(templ_low,bound,m_cand-min_m,0) == 1
                    m_high=m_cand;
                else
                    m_low=m_cand;
                end
                delta=m_high-m_low;
            end
            b(i)=m_cand;
        end
    end

    end

%
    ph_out=min_p:step_p:max_p;
%
%----- End of hr_bnds1.m -----
%
function flag=chk_span(fr_p,bound,i_mag,i_ph)
%
% Function determines whether closed-loop span (in db's) of
% input uncertainty template translated by given magnitude
% and phase in the Nichols chart, lies within specified
% M-circle bound.
%
% fr_p : Uncertainty template (row vector of complex numbers)
%         on Nichols chart. Real part is phase in degrees,
%         imaginary part is 20*log10(magnitude)
% bound : Gain span (db) that closed-loop must satisfy for all
%         elements of uncertainty template
% i_mag : Magnitude translation interval in db's
% i_ph : Phase translation interval in degrees
% flag : 1 if bound is satisfied, 0 otherwise
%-----
% Error messages:
    if size(fr_p)*[1;0] ~= 1
        error('Template should be row vector ...');
    end
%-----
    n_templ=size(fr_p)*[0;1]; % number of points in uncertainty templates
    templ1=fr_p+(i_ph+sqrt(-1)*i_mag).*ones(1,n_templ); % translate
%
% convert to polar form
    magnitude=(10.*ones(1,n_templ)).^(imag(templ1)./20);
    phase=real(templ1);
    templ3=p2r(magnitude,phase); % polar to rectangular conversion
    m_val=zeros(1,n_templ); % initialize vector for m-values of points

```

```

% in template
%
for i=1:n_templ
    m_val(i)=(real(templ3(i)))^2+(imag(templ3(i)))^2;
    m_val(i)=m_val(i)/((real(templ3(i))+1)^2+(imag(templ3(i)))^2);
    m_val(i)=sqrt(m_val(i));
end
%
m_val_max=max(m_val); % maximum m-circle touching template
m_val_min=min(m_val); % minimum m-circle touching template
m_span=20*log10(m_val_max/m_val_min); % max template span in db
%
if m_span <= bound
    flag=1;
else
    flag=0;
end
%
%----- End of chk_span.m -----

```

### 9.3.6 M.file to calculate disturbance bounds

```

function [b,ph_out]=dis_bnds(fr_p,ref_p,min_m,max_m,min_p,max_p,...
ph_ps,bound,tol_m)
%
% Function determines the Horowitz disturbance-rejection regions
% (minimum gain regions in Nichols chart at which template
% sensitivity variations lie (just) under specified disturbance
% attenuation bound. Function uses bisection algorithm
% for given phase span and for specified magnitude tolerance.
%
% fr_p : Uncertainty template (row vector of complex numbers)
%        on Nichols chart. Real part describes the phase in
%        degrees, imaginary part is equal to 20*log10(magnitude).
% ref_p : Reference point on which translation is based (could
%        be arbitrary point inside template, first point in
%        template etc.)
% min_m : Minimum magnitude open-loop level in db
% max_m : Maximum magnitude open-loop level in db
% min_p : Minimum open-loop phase in degrees (typically -180)
% max_p : Maximum open-loop phase in degrees (typically 0)
% ph_ps : Number of phase points linearly spaced between min
%        and max values (>=2)
% bound : Disturbance attenuation level (db) that sensitivities of all
%        points of uncertainty template must satisfy
% tol_m : Magnitude contour tolerance (db)
% b      : Horowitz disturbance contour (db)
% ph_out: (Linearly-spaced) phase vector corresponding to b (degrees)
%
%-----
% Error messages:
%

```

```

if size(fr_p)*[1;0] ~= 1
    error('First variable should be row vector ...');
end
if min_m >= max_m
    error('Minimum magnitude level should not exceed maximum level ...');
end
if min_p >= max_p
    error('Minimum phase level should not exceed maximum level ...');
end
if tol_m <= 0
    error('Magnitude tolerance variable should be positive ...');
end
%-----
%
step_p=(max_p-min_p)/(ph_ps-1); % phase step
mag_ref=imag(ref_p); % magnitude of reference point in db's
ph_ref=real(ref_p); % phase of reference points in db's
b=zeros(1,ph_ps); % initialise Horowitz contour
n_tmpl=size(fr_p)*[0;1]; % number of points in uncertainty templates
max_iter=100; % maximum number of iterations
%
for i=1:ph_ps
%
    disp(['Phase point # ',int2str(i)]);
    phase=min_p+(i-1)*step_p;
%
    translate:
    templ_low=fr_p+ ...
    (phase-ph_ref+sqrt(-1)*(min_m-mag_ref)).*ones(1,n_tmpl);
    templ_high= ...
    fr_p+(phase-ph_ref+sqrt(-1)*(max_m-mag_ref)).*ones(1,n_tmpl);
%
    if chk_dist(templ_low,bound,0,0) == 1
        b(i)=min_m;
    elseif chk_dist(templ_high,bound,0,0) == 0
        b(i)=max_m;
    else
        delta=max_m-min_m;
        m_high=max_m;
        m_low=min_m;
        n_iter=1;
        while delta > tol_m & n_iter <= max_iter
            n_iter=n_iter+1;
            m_cand=0.5*(m_high+m_low);
            if chk_dist(templ_low,bound,m_cand-min_m,0) == 1
                m_high=m_cand;
            else
                m_low=m_cand;
            end
            delta=m_high-m_low;
        end
        b(i)=m_cand;
    end
end
end

```

```

    end
%
    ph_out=min_p:step_p:max_p;
%
%----- End of dis_bnds.m -----
%
    function flag=chk_dist(fr_p,level,i_mag,i_ph)
%
% Function determines whether the maximum sensitivity of
% input uncertainty template translated by given magnitude
% and phase in the Nichols chart, is less than a specified
% level (in db's).
%
% fr_p : Uncertainty template (row vector of complex numbers)
%        on Nichols chart. Real part is phase in degrees,
%        imaginary part is 20*log10(magnitude)
% level : value in (db) which must overbound the sensitivities
%        of all members of uncertainty template
% i_mag : Magnitude translation interval in db's
% i_ph  : Phase translation interval in degrees
% flag  : 1 if bound is satisfied, 0 otherwise
%
%-----
% Error messages:
%
    if size(fr_p)*[1;0] ~= 1
        error('Template should be row vector ...');
    end
%-----
    n_templ=size(fr_p)*[0;1]; % number of points in uncertainty templates
    templ1=fr_p+(i_ph+sqrt(-1)*i_mag).*ones(1,n_templ); % translate
%
% convert to polar form
    magnitude=(10.*ones(1,n_templ)).^(imag(templ1)./20);
    phase=real(templ1);
    templ3=p2r(magnitude,phase); % polar to rectangular conversion
    s_val=zeros(1,n_templ); % initialise vector of sensitivities
%
    for i=1:n_templ
        s_val(i)=1/((real(templ3(i))+1)^2+(imag(templ3(i)))^2);
        s_val(i)=sqrt(s_val(i));
    end
%
    s_val_max=max(s_val); % maximum sensitivity
    s_val_max=20*log10(s_val_max); % back to db's
%
    if s_val_max <= level
        flag=1;
    else
        flag=0;
    end
%

```

%----- End of chk\_dist.m -----

### 9.3.7 M-file used to construct $M$ -contours in the Nichol's chart

```
function u_temp=m_cir(m_value,phi,tol)
% m_cir.m
% Create m-circle for given phase vector (phi)
%
% input variables:
%
% phi:      row vector of phases (deg) between -360 and 0
% m_value:  M (linear)
% tol:      if given M circle does not pass through constant
%           phase line (phi(i)) all elements of ith column of
%           u_temp are set to tol (usually -80db).
%
% output variables:
%
% u_temp:   matrix of n rows and p columns where p=size(phi,2)
%           if m_value>1, n=2, else n=1. If constant phi(i) line
%           intersects the m-circle, then u_temp(i,:) contain the
%           co-ordinate(s) of the corresponding point(s) of the circle,
%           with the point of largest magnitude (if more than one) in
%           the first row; else u_temp(i,j)=tol for i=1,2.
%
nphi=size(phi,2);
%
if m_value > 1
    u_temp=zeros(2,nphi); % create matrix to store co-ordinates (NC)
    for i=1:nphi
        if (phi(i) <= 0 & phi(i) >= -90) | (-360 <= phi(i) & phi(i) <= -270 )
            u_temp(1,i)=tol;
            u_temp(2,i)=tol;
        else
            slope=tan(pi+phi(i)*pi/180);
            if abs(slope) > 1/sqrt(m_value*m_value-1)
                u_temp(1,i)=tol;
                u_temp(2,i)=tol;
            else
                x_var=[(1+slope^2) -2*(m_value^2)/(1-m_value^2) ...
                    (-m_value^2)/(1-m_value^2)];
                root12=roots(x_var);
                x_1=min(root12);y_1=slope*x_1;
                x_2=max(root12);y_2=slope*x_2;
                u_temp(1,i)=10*log10(x_1*x_1+y_1*y_1);
                u_temp(2,i)=10*log10(x_2*x_2+y_2*y_2);
            end
        end
    end
elseif m_value==1
    u_temp=zeros(2,nphi); % create matrix to store co-ordinates (NC)
```

```

    for i=1:nphi
        if (phi(i) <= 0 & phi(i) >= -90) | (-360 <= phi(i) & phi(i) <= -270 )
            u_temp(:,i)=[tol;tol];
        else
            slope=tan(pi+phi(i)*pi/180);
            x=-.5;
            y=slope*x;
            u_temp(:,i)=[10*log10(x*x+y*y);10*log10(x*x+y*y)];
        end
    end
end
else
    u_temp=zeros(2,nphi); % create matrix to store co-ordinates (NC)
    for i=1:nphi
        if phi(i) == 0 | phi(i) == -360
            x=m_value/(1-m_value);
            u_temp(:,i)=[20*log10(x);20*log10(x)];
        elseif phi(i) == -90 | phi(i) == -270
            y=m_value*m_value/(1-m_value*m_value);
            u_temp(:,i)=[10*log10(y);10*log10(y)];
        elseif phi(i) == -180
            x=m_value/(1+m_value);
            u_temp(:,i)=[20*log10(x);20*log10(x)];
        else
            slope=tan(pi+phi(i)*pi/180);
            x_var=[(1+slope^2) -2*(m_value^2)/(1-m_value^2) ...
                (-m_value^2)/(1-m_value^2)];
            root12=roots(x_var);
            if phi(i) > -90 | phi(i) < -270
                x=max(root12);
                y=slope*x;
                u_temp(:,i)=[10*log10(x*x+y*y);10*log10(x*x+y*y)];
            else
                x=min(root12);
                y=slope*x;
                u_temp(:,i)=[10*log10(x*x+y*y);10*log10(x*x+y*y)];
            end
        end
    end
end
end
%----- end of m_cir.m -----

```

### 9.3.8 M-file to construct high-frequency bounds

```

function cont = hf_bound(m_cir,max_inf,min_inf,nom_inf)
%
% Function computes the single high frequency boundary for which
% closed-loop variations are bounded by specified level (so that
% boundary should include the M-circle corresponding to that
% level) on the Nicols chart. Uncertainty at high frequencies
% is described by vertical line on the chart specified
% by three values, the maximum, minimum and nominal (relative)
% levels.

```

```

%
% max_inf : Maximum (relative) level of uncertainty template
%           at infinity (db)
% min_inf : Minimum (relative) level of uncertainty template
%           at infinity (db)
% nom_inf : Nominal (relative) level of uncertainty template
%           at infinity (db)
% m_cir    : Closed loop magnitude bound (db); m_circ should be
%           positive to produce closed contour on Nichols chart.
% cont     : Complex vector of calculated high-frequency contour.
%           Real part is phase in degrees. Imaginary part is
%           magnitude in db's.
%-----
% Error messages:
%
%   if m_cir <= 0
%       error('Positive value required for CL magnitude value ...');
%   end
%
%   if max_inf < nom_inf | nom_inf < min_inf | max_inf < min_inf
%       error('Incorrect ordering of uncertainty-template parameters ...');
%   end
%
%   if max_inf < 0 | nom_inf < 0 | min_inf < 0
%       error('Uncertainty-template parameters should be positive ...');
%   end
%-----
% Test data:
% max_inf=20; % maximum level
% min_inf=10; % minimum level
% nom_inf=0;  % nominal level
% m_cir=2;    % M-circle value
%-----
dp1_inf=max_inf-nom_inf; % upper spread of uncertainty template
%                         % at infinity
dp2_inf=nom_inf-min_inf; % lower spread of uncertainty template
%                         % at infinity
m_circ_f=mcirc(m_cir); % required M-circle in rectangular
%                     % coordinates
n_points=max(size(m_circ_f)); % number of points in M-circle
[mag_mc,ph_mc]=r2p(m_circ_f); % rectangular to polar conversion
mag_mc=20.*log10(mag_mc); % transform magnitude for Nicols chart
ph_mc=fixphase(ph_mc); % transform phase
%
% get maximum and minimum phases for ph_mc corresponding to the
% required m-circle
[ph_p_min,ind_p_min]=min(ph_mc);
[ph_p_max,ind_p_max]=max(ph_mc);
%
[mag_m_min,ind_m_min]=min(mag_mc); % index of minimum db level
%                               % on m_circle
low_m=ph_mc(ind_p_max:ind_p_min)+ ...

```

```

    sqrt(-1).*mag_mc(ind_p_max:ind_p_min); %low-part of circle
    up_m=ph_mc(ind_p_min:n_points)+ ...
    sqrt(-1).*mag_mc(ind_p_min:n_points);
    up_m=[up_m ; ph_mc(1:ind_p_max)+sqrt(-1).*mag_mc(1:ind_p_max)];
%
% translate low-part of circle by dp1_inf downwards:
    cont1=-sqrt(-1)*dp1_inf.*ones(ind_p_min-ind_p_max+1,1)+low_m;

% translate upper-part of circle by dp2_inf upwards:
    cont2=sqrt(-1)*dp2_inf.*ones(max(size(up_m)),1)+up_m;

% Final contour:
    cont=[cont1 ; cont2]; % combine contours
    cont=[cont ; cont(1) ]; % include last point for closed graphs in plots
%
%----- end of hf_bound.m -----

```

### 9.3.9 M-file to check Nichols stability criterion

```

function flag=check_nichols1(num,den,freq,tol,tol1,verb)
%
% Checks the stability of an open-loop system from its frequency response
% in Nichols chart; Function returns the value of flag based on closed-loop
% stability, i.e. flag=1 means closed-loop stable, flag=0 means closed-loop
% is unstable. Input variables:
%
% num,den: System transfer function numerator/denominator polynomials.
% freq    : User specified frequency-array (row-vector) in rads/s.
% tol     : denom coefficients are perturbed by tol so that generically
%           there are no jw-axis poles.
%
n=size(den,2); % size of den
nfreq=size(freq,2); % no of frequencies;
den_p=tol1*ones(1,n); % perturbation in denominator's coefficients
den=den+den_p; % perturbed denominator
no_unst_poles=sum(real(roots(den))>=0);
[m,p]=nichols(num,den,freq); % m and p are columns!
nic=p+20*j*log(m);
nic=renorm(nic); % ensure phase between -360 and 0 deg
%
distance=min(abs(nic+180*ones(nfreq,1)));
if distance < tol
    flag = 0; % if almost through -1 point cl is unstable
    return
end
%
ph=real(nic); % deg
mag=imag(nic); % dB's
%
index_mag_g0=find(mag>0);
mag1=mag(index_mag_g0); % remove mags < 0 dB
ph1=ph(index_mag_g0); % and corresponding phases

```



```

freq1=freq(index_mag_g0); % remove mags < 0 dB
%
index_ph=(ph1>-180); % phase indices to the right of -180 line
if isempty(index_ph)
    index3=0; % all phases to the left of -180 line
else
    stab_index1=diff(index_ph); % phase differences
    stab_index2=find(abs(stab_index1)==1); % indices where transitions occur
    %ph_transitions=ph1(stab_index2); % phases where transitions occur
%
    for i=1:size(stab_index2,1)
        if abs(ph1(stab_index2(i))-ph1(stab_index2(i)+1)) > 300 % phase disc.
            stab_index1(stab_index2(i))=0;
        end
    end
    index3=sum(stab_index1);
end
%
if index3 == no_unst_poles
    flag=1; % CL stable
else
    flag=0; % CL unstable
end
%
if verb==1
    figure(1)
    plot(nic)
    grid
end
%
%----- end of check_nichols1.m -----

```

### 9.3.10 M-file to design the optimum PID controller

```

function [kd_opt,ki_opt,kp_opt]=pid_opt(w,bounds,phi,convex_temp,u_contour)
% pid_opt.m
% Design optimal PID controller subject to QFT bounds
% Inputs:
% w      : frequencies of interest
% bounds: overall bounds (horowitz+disturbance+hfc)
% phi    : phase range
% convex_temp: plant uncertainty templates
% u_contour: HFC (input only the first row)
%
% Outputs:
% kd_opt,ki_opt,kp_opt: optimal kd and corresponding ki and kp parameters
%
%-----%
% Find the phase of the nominal plant for the first two frequencies
nw=size(w,2); % find size of array of frequencies
nom_p=convex_temp(:,1); % nominal plant
nom_p_phases=real(nom_p); % overall phases of nominal plant

```

```

nom_p_mags=imag(nom_p); % overall magnitudes of nominal plant in dB
% Set range for the first fix phase for the first frequency w(1)
lower1=-90+nom_p_phases(1); % lower limit
low_phi_diff=abs(lower1-phi); % find differences
[low_limit,index_low]=min(low_phi_diff); % minimum difference
upper1= 90+nom_p_phases(1); % upper limit
if upper1 > 0 | upper1 == 0 % check if upper limit exceeds 0 degrees
    upper1=0; % set upper1 to zero
end
up_phi_diff=abs(upper1-phi); % differences between upper and phi_range
[up_limit,index_up]=min(up_phi_diff); % minimum difference
n1=size(phi(index_low:1:index_up),2); % find no of iterations
phi_i=linspace(lower1,upper1,n1);
% Set range for the second fix phase for the second frequency w(8)
% Similarly.....
rej_gain=u_contour==-80; % set all values ==-80 to 1 (all others to 0)
index_rej_value=find(rej_gain); % find indices of the non-zero elements
lower_last=phi(index_rej_value(1)); % lower limit of phase
low_phi_diff2=abs(lower_last-phi); % differences between lower and
%phi_range
[low_limit2,index_low2]=min(low_phi_diff2); % minimum difference
upper_last= 90+nom_p_phases(nw); % upper limit
if upper_last > 0 | upper_last == 0 % check if upper limit exceeds
%0 degrees
    upper_last=0; % set upper_last to zero
end
up_phi_diff2=abs(upper_last-phi); % differences between upper and
%phi_range
[up_limit2,index_up2]=min(up_phi_diff2); % minimum difference
n2=size(phi(index_low2:1:index_up2),2); % find no of iterations
phi_j=linspace(lower_last,upper_last,n2);
% Find sizes of phi_i and phi_j
nphi_i=size(phi_i,2);
nphi_j=size(phi_j,2);
% Perform SVD of matrix A for all combinations of phi_i and phi_j
kd_temp=[];
ki_temp=kd_temp;
kp_temp=kd_temp;
kd_star=[]; % initialise derivative constant array
ki_star=kd_star; % initialise integral constant array, same size as kd
kp_star=kd_star; % initialise proportional constant array, same size as kd
for i=1:nphi_i
    for r=1:nphi_j
        A=[1 -1/(w(1)^2) -tan(phi_i(i)*pi/180)/w(1);
            1 -1/(w(nw)^2) -tan(phi_j(r)*pi/180)/w(nw)];
        [U,S,V]=svd(A); % SVD for the case A=2x3
        V2=V(:,3)'; % holds kd,ki,kp without the gain lambda
        if (V2(1)>0 & V2(2)>0 & V2(3)>0)|(V2(1)<0 & V2(2)<0 & V2(3)<0)
            % find phase of nominal open loop for the rest of the frequencies
            % first include the two first frequencies and the relevant
            %fixed phases
            psi=zeros(1,nw); % re-initialize

```

```

pow1=[]; % initialize power
for c=1:nw
    if c==1
        psi(c)=phi_i(i);
    elseif c==nw
        psi(c)=phi_j(r);
    else
        psi(c)=nom_p_phases(c)+...
            (atan(((V2(1)*w(c))-(V2(2)/w(c)))/V2(3))*180/pi);
        % phase of nominal open loop for remaining frequencies
        % according to fixed phases for the first two frequencies
    end
    % Difference between psi and phases of bounds
    diff2=psi(c)-phi;
    [phase2,index2]=min(abs(diff2));
    % Find gain lambda for each case
    var1=10*log10(V2(3)^2+(V2(1)*w(c)-(V2(2)/w(c)))^2); % dummy
    % variable to things easier when proceeding to the next line
    pow11=bounds(c,index2)-nom_p_mags(c)-var1;
    pow1=[pow1,pow11];
end
[pow21,index3]=max(abs(pow1)); % find maximum for this combination
% of phases
kp_temp1=10^(pow21/20)*abs(V2(3)); %
kd_temp1=sign(V2(3))*10^(pow21/20)*(V2(1)); % minimum kd to achieve
% specs
ki_temp1=sign(V2(3))*10^(pow21/20)*(V2(2));
else
    ki_temp1=inf;
    kd_temp1=inf;
    kp_temp1=inf;
end
kp_temp=[kp_temp,kp_temp1];
kd_temp=[kd_temp,kd_temp1];
ki_temp=[ki_temp,ki_temp1];
end
[kd1,index4]=min(abs(kd_temp)); % find the minimum of the minima of above
% combinations
kd_star=[kd_star,kd1]; % store the current optimal kd
ki_star=[ki_star,ki_temp(index4)];
kp_star=[kp_star,kp_temp(index4)];
end
[kd_opt,index5]=min(abs(kd_star)); %the optimal kd
ki_opt=ki_star(index5);
kp_opt=kp_star(index5);
%
%-----end of pid_opt.m-----

```

### 9.3.11 M-File for graphical design of the controller

```

%
% Graphdes.m - Graphical design for main design

```

```

% Graphical design using phase lead and phase lag controllers in cascade.
%-----%
% Step 1: Initial procedure (settings,specs,unc. templates,nominal open loop
% before controller,u-contour,hor. templates)
%-----%
clear all;
close all;
create;
% Define the range of frequencies of interest & of general frequency range
phi_new=[-360:1:0]; % for illustration purposes
% Plot convex hull uncertainty templates figure 1
m_grid1(-270,0,40,m_circ_range)
title('Uncertainty templates for the range of chosen frequencies')
plot(c_convex.','k')
plot(c_convex.','k+')
plot(nom_ol_mark,'ko'); % nominal plant for marked frequencies
nom_ol_mark_text=nom_ol_mark+3;
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
% Plot the freq. response of the dist. rej model figure 2
figure % create figure for the disturbance frequency response
semilogx(w_range,20*log10(mag_dist),'k-',w_interest_dis,...
20*log10(mag_dist_mark),'ko');
grid;
coordy=[20*log10(mag_dist_mark)+3];
coordx=w_interest_dis;
for i=1:size(w_interest_dis,2)
    text(coordx(i),coordy(i),num2str(w_interest_dis(i)))
end
title ('2nd order disturbance rejection model frequency reponse');
xlabel('ang. freq. rads/s');
ylabel('magnitude dB');
% Nichols chart including all details 3
m_grid1(-270,0,40,m_circ_range)
title('UHFB, Hor & Dist Bounds, Nominal OL')
plot(nom_ol,'k--');
plot(nom_ol_mark,'ko');
plot(phi,b_hor.','k-');plot(phid,b_dis.','k-.');
plot(hfc,'k-')
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
% Nichols chart including overall bounds,hfc,m_circle figure 4
m_grid1(-270,0,40,m_circ_range)
title('Overall bounds and nominal ol')
plot(nom_ol,'k--');
plot(nom_ol_mark,'ko');

```

```

        coordy=imag(nom_ol_mark_text);
        coordx=real(nom_ol_mark_text);
        for i=1:size(w_interest_track,2)
            text(coordx(i),coordy(i),num2str(w_interest_track(i)))
        end
        plot(phi,b_all.','k-');
        plot(hfc,'k-')
% Take in account the fact that some templates may include points at -80 dBs
%-----%
% Step 2: Phase lead/lag Controller design using graphical design
%-----%
% Set vars y=Y=1 & n=N=0
y=1; Y=y;
n=0; N=n;
nk=1;dk=1;
nkstore=[0 1]; % initialise matrix to store numerators of controllers
dkstore=[0 1]; % initialise matrix to store denominators of controllers
controller=[0 1 0 1]; % initialise matrix to store controllers
nol=num_p;dol=den_p;
%
m_grid1(-270,0,40,m_circ_range)
plot(nom_ol,'k--');
plot(nom_ol_mark,'ko');
plot(phi,b_all.','k-');
nom_ol_mark_text=nom_ol_mark+3;
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
plot(hfc,'k-')
% Loop of designing the controller
%
% Ask before starting controller design
question_1=input('Continue with controller design? [y/n] > ');
while question_1==1
% proceed to feedback controller design
% -----
[nk,dk,w_click]=gr_des(nol,dol,w_range);
nkstore=[nkstore;nk];
dkstore=[dkstore;dk];
controller=[controller;nk dk];
row_contr=size(controller,1);% update (row) size of controller matrix
row_nkstore=size(nkstore,1); % update size of the numerator matrix of
% the controller
row_dkstore=size(dkstore,1); % update size of denominator matrix of
% the controller
%
[nol,dol]=series(nol,dol,nk,dk);
[mol,pol]=bode(nol,dol,w_range);
[mol_m,pol_m]=bode(nol,dol,w_interest_track);
[m_mark,p_mark]=bode(nol,dol,w_click);

```

```

%
plot(phi,b_all.','k-');
plot(hfc,'k-');
plot(pol+sqrt(-1)*20*log10(mol),'k-');
plot(pol_m+20*sqrt(-1)*log10(mol_m),'kdiamond');
coordy=imag(20*sqrt(-1)*log10(mol_m));
coor dx=pol_m+3;
for i=1:size(w_interest_track,2)
    text(coor dx(i),coordy(i),num2str(w_interest_track(i)))
end
plot(p_mark+20*sqrt(-1)*log10(m_mark),'k*');
%
% Before continuing design:
question_2=input('Keep the controller? [y/n] > ');
if question_2==1
    close,m_grid1(-270,0,40,m_circ_range)
    plot(phi,b_all.','k-');
    plot(hfc,'k-');
    plot(pol+sqrt(-1)*20*log10(mol),'k--');
    plot(pol_m+20*sqrt(-1)*log10(mol_m),'ko');
    coordy=imag(20*sqrt(-1)*log10(mol_m));
    coor dx=pol_m+3;
    for i=1:size(w_interest_track,2)
        text(coor dx(i),coordy(i),num2str(w_interest_track(i)))
    end
    plot(p_mark+20*sqrt(-1)*log10(m_mark),'k*');
    question_1=input('Continue with controller design? [y/n] > ');
elseif question_2==0
%
    nkstore(row_nkstore,:)=[]; % delete numerator of controller
    dkstore(row_dkstore,:)=[]; % delete denominator of controller
    controller(row_contr,:)=[]; % delete the compensator
    [rown,coln]=size(nkstore);
    [rowd,cold]=size(dkstore);
    [nol,dol]=series(num_p,den_p,nkstore(1,:),dkstore(1,:));
    if rown>1
        for condition=2:rown
            [nol,dol]=series(nol,dol,nkstore(condition,:),...
                dkstore(condition,:));
        end
    end
    [mol,pol]=bode(nol,dol,w_range);
    [mol_m,pol_m]=bode(nol,dol,w_interest_track);
    [m_mark,p_mark]=bode(nol,dol,w_click);
    close,m_grid1(-270,0,40,m_circ_range)
    plot(phi,b_all.','k-');
    plot(hfc,'k-');
    plot(pol+sqrt(-1)*20*log10(mol),'k--');
    plot(pol_m+20*sqrt(-1)*log10(mol_m),'ko');
    coordy=imag(20*sqrt(-1)*log10(mol_m));
    coor dx=pol_m+3;
    for i=1:size(w_interest_track,2)

```

```

        text(coordx(i),coordy(i),num2str(w_interest_track(i)))
    end
    plot(p_mark+20*sqrt(-1)*log10(m_mark),'kx');
    question_1=input('Continue with controller design? [y/n] > ');
end
end
if question_1==0
    disp('End of design')
end
%
% adding designed OL in figure
% Find overall OL
[num_controller,den_controller]=coseries(controller);
[num_ov,den_ov]=series(num_p,den_p,num_controller,den_controller);
[mag_ol,phase_ol]=bode(num_ov,den_ov,logspace(-0.6,3,100));
mag_ol=20*log10(mag_ol); % in dBs
[mag_ol_m,phase_ol_m]=bode(num_ov,den_ov,w_interest_track);
mag_ol_m=20*log10(mag_ol_m); % in dBs
designed_ol=phase_ol+(sqrt(-1)*mag_ol);
designed_ol_m=phase_ol_m+(sqrt(-1)*mag_ol_m); % marked frequencies
close,m_grid1(-270,0,40,m_circ_range)
plot(phi,b_all.','k-');
plot(hfc,'k-');
plot(nom_ol,'k--');
plot(nom_ol_mark,'ko');
coordy=imag(nom_ol_mark_text);
coordx=real(nom_ol_mark_text);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
plot(designed_ol,'k-')
plot(designed_ol_m,'ko')
coordx=real(designed_ol_m+3);
coordy=imag(designed_ol_m);
for i=1:size(w_interest_track,2)
    text(coordx(i),coordy(i),num2str(w_interest_track(i)))
end
title('Nominal OL & Desinged OL')
% ------%
% Step 3: CL with Pre-filter (2nd order)
% ------%
% Plot specifications, figure 5
figure
semilogx(w_range,mu,'k-',w_range,ml,'k-',w_interest_track,mu_m,...
'ko',w_interest_track,ml_m,'ko')
grid,hold
% Settings:
A=1;
k_r=[1 5 8 10];
a_r=[1 5 8 10];
wc1=3.5; % 1st cut-off
wc2=7.8; % 2nd cut-off

```

```

% Frequency response at all frequencies
[magsys,phasesys]=frcl_ll(A,wc1,wc2,k_r,a_r,num_controller,...
    den_controller,w_range);
% Frequency response at marked frequencies
[magsys_m,phasesys_m]=frcl_ll(A,wc1,wc2,k_r,a_r,num_controller,...
    den_controller, w_interest_track);
% Add the new information in figure 5
semilogx(w_range,magsys,'k--',w_interest_track,magsys_m,'ko')
title('Frequency response of cl designed system')
xlabel('ang. frequency rads/s');
ylabel('magnitude dB');
% Frequency response of Tyd at all frequencies
[magsysd,phasesysd]=disr_fr2(k_r,a_r,num_controller,den_controller,w_range);
% Frequency response of Tyd at marked frequencies
[magsys_dm,phasesys_dm]=disr_fr2(k_r,a_r,num_controller,den_controller,...
    w_interest_dis);
% Add the new information in figure 6 with the disturbance spec.
figure
semilogx(w_range,20*log10(mag_dist),'k-',...
    w_interest_dis,20*log10(mag_dist_mark),'ko');
grid;
title('Rejection of the disturbance in the frequency domain');
xlabel('ang. freq. rads/s');
ylabel('magnitude dB');hold;
semilogx(w_range,magsysd,'k--',w_interest_dis,magsys_dm,'ko')
% Find the differences of the bounds at the given freqs
difference_tracking=zeros(size(magsys_m,1),1);
dis_gain=zeros(size(magsys_dm,1),1);
for i=1:size(magsys_m,1)
    max_limit=max(magsys_m(i,:));
    min_limit=min(magsys_m(i,:));
    difference_tracking(i)=max_limit-min_limit;
end
for i=1:size(magsys_dm,1)
    max_limit=max(magsys_dm(i,:));
    dis_gain(i)=max_limit;
end
% Now the step responses...
t=[0:1000]/200; % simulation time (5 secs)
y_upper=step(num_u,den_u,t); % specifications
y_lower=step(num_l,den_l,t); % specifications
% System designed
[stepall,tsim]=stepsim2(k_r,a_r,num_controller,den_controller,...
    1,wc1,wc2,t);
% Plot responses
figure
grid,hold
title('Step responses of designed cl system and specifications');
xlabel('Simulation time secs');
ylabel('Amplitude');
plot(t,y_upper,'k-',t,y_lower,'k-',t,stepall,'k--');
% Disturbance rejection in the time domain

```



```

% Derived model..
d_rej_model=step(num_model,den_model,t);
% Designed system
dist_resp=distrej2(k_r,a_r,num_controller,den_controller,t);
% Plot result
figure
grid,hold
title('Disturbance rejection in time domain');
xlabel('Simulation time secs');
ylabel('Amplitude');
plot(t,d_rej_model,'k-.',t,dist_resp,'k-');
%
%----- end of graphdes.m -----
%
% create.m - Produce data used for graphdes.m
%
clear all; % clear workspace
%
% Define the range of frequencies of interest & of general frequency range
m_circ_range=[0.0001 0.01 0.1 0.5 0.7 1 1.4 2.8];
w_interest_track=[.5 1 2 3 5 10 40 80]; % for illustration purposes
m_circ_value=1.2; % m_circle for UHFB
%
% For disturbance rejection low frequency range needed
up_limit_w_dis=80; % upper limit of freq. for disturbance rejection
trans1=w_interest_track <= up_limit_w_dis;
index_dis=find(trans1);
w_interest_dis=w_interest_track(index_dis); %dist. rej. freq. of interest
w_range=logspace(-2,2.5,100); % frequency range
nw_mark=size(w_interest_track,2); % size of frequency range
%
% Define plant uncertainty
k=[1:1:10];
a=[1:1:10];
num_p=[0 0 1]; % nominal plant's numerator
den_p=[1 1 0]; % nominal plant's denominator
%
% Define input specifications:
num_u=.6585*[1 30]; %numerator of upper model
den_u=conv([1 2+sqrt(-1)*3.969],[1 2-sqrt(-1)*3.969]);
den_u=real(den_u); % denominator of upper model
num_l=8400; % numerator of lower model
den_l=poly([-3 -4 -10 -70]); % denominator of lower model
%
% Derive the frequency response of the upper and lower models
% ...for the general range of frequencies
[mu,pu]=bode(num_u,den_u,w_range);
mu=20*log10(mu); % translation in dBs (*) upper bound
[m1,p1]=bode(num_l,den_l,w_range);
m1=20*log10(m1); % (*)lower bound
% ...for the frequencies of interest
[mu_m,pu_m]=bode(num_u,den_u,w_interest_track);

```

```

mu_m=20*log10(mu_m); % (*)upper bound
[ml_m,pl_m]=bode(num_l,den_l,w_interest_track);
ml_m=20*log10(ml_m); % (*) lower bound
spec=mu_m-ml_m; % tracking specs
% 'mark' indicates point at the frequencies of interest
%
% Derive the uncertainty templates (use convex hull to
% reduce the computational burden of the algorithm)
c_initial=temp_cs1(k',a',w_interest_track'); % uncertainty templates
c_convex=c_hull1(c_initial); % convex hull
% Nominal open loop
nom_ol=temp_cs1(1,1,w_range');
nom_ol_mark=temp_cs1(1,1,w_interest_track');
%
% Introducing 2nd order disturbance rejection
%
do=1; % step input disturbance
ap=.1; % maximum allowed gain that dist. can reach after ts
tx=2; % settling time for the dist. input in sec
pu=25; % max. percent overshoot of the dist. input
[a,b,num_model,den_model] = dop_mod2(do,ap,tx,pu);
%
% The disturbance rejection is translated in the frequency domain
[mag_dist,phase_dist]=bode(num_model,den_model,w_range); % storing
%the coordinates of he bode plot
% storing the coordinates of the mark frequencies of the bode plot
[mag_dist_mark,phase_dist_mark]=bode(num_model,den_model,w_interest_dis);
%
% Finding the maximum rejection bound (db)
bound_max=(20*log10(mag_dist_mark));
%
b_dis=[];
for i=1:size(w_interest_dis,2)
    disp(['calculating dis. temp. for frequency ',int2str(i)]);
    [bd,phid]=dis_bnds(c_convex(i,:),c_convex(i,1),-80,80,-180,0,...
    181,bound_max(i),.1);
    b_dis=[b_dis;bd];
end
%
% Find Horowitz templates and U-contour
b_hor=[]; % Horowitz templates creation
for i=1:size(w_interest_track,2)
    disp(['Calculating Horowitz templates for frequency ',int2str(i)]);
    [b,phi]=hr_bnds1(c_convex(i,:),c_convex(i,1),-80,80,-180,0,181,...
    spec(i),.1);
    b_hor=[b_hor;b];
end
%
% HFC (High Frequency Contour) M=m_circ_value and m-circle m_circ_value
nphi=size(phi,2); % size of number of phases
u_tmpl=zeros(nw_mark,nphi); % initialise template for m-circle
for i=1:nw_mark

```

```

        ut=m_cir(m_circ_value,phi,-80); % m-circle m_circ_value
        ut=ut(1,:);
        u_templ(i,:)=ut;
    end
    hfc=hf_bound(20*log10(m_circ_value),40,0,0); % hfc
%
% Find overall bounds, nominal open loop (for general range of frequencies
% and for marked frequencies)
    b_all1=zeros(size(b_dis)); % Merged templates
    b_all=zeros(size(b_hor)); % Overall templates
    for i=1:size(b_hor,1)
        if i <= size(b_dis,1)
            b_all1(i,:)=max(b_hor(i,:),b_dis(i,:));
            b_all(i,:)=max(b_all1(i,:),u_templ(i,:));
        else
            b_all(i,:)=max(b_hor(i,:),u_templ(i,:));
        end
    end
end
%
%----- end of create.m -----
%
function [nk,dk,w_click]=gr_des(sys,w)
%
% Visual design of the compensator (phase lead/lag)
% The user clicks using the mouse on a position of the nominal plant
% and then clicks on the position (s)he wants to move the graph.
% Phase lead, phase lag or constant gain can be added in order to achieve
% a loop shaping which will satisfy that the closed loop performance will
% be the desired.
%
% INPUTS:
% sys: OL system
% w: frequency row-array
%
% OUTPUTS:
% nk,dk: designed controller numerator and denominator
% w_click: frequency which was chosen
%
[m,p]=bode(sys,w); % bode plot coordinates mag. and phase
m=m(1,:);p=p(1,:);m=20*log10(m);p=mod(p,-360);
%
while 1
    [ph,mag]=ginput(2); % use the mouse to pick point where max
    % lead/lag is injected
    point1=[ph(1) mag(1)]; % phase-mag of point 1
    point2=[ph(2) mag(2)]; % phase-mag of point 2
    phase_diff=ph(2)-ph(1); % find the phase difference between
    % the two points
    if abs(phase_diff) < 90 % allowable range for phase-lead/lag
    % design is -90 to +90 degrees
        break;break;
    end
end

```

```

end
%
mag_diff=mag(2)-mag(1); % find the mag. difference between the
% two points
nf=size(w,2); % no of frequencies
test1=(ph(1)+sqrt(-1)*mag(1))*ones(1,nf); % test array1
test2=p+sqrt(-1)*m; % frequency response
test3=abs(test2-test1); % error
[err1,index]=min(test3); % select frequency so that absolute
% sum of gain and phase error is minimum
w_click=w(index); % this is the frequency chosen to move
%
if phase_diff > 0 % phase-lead controller design
    [nk,dk]=ph_lead(w_click,phase_diff,10^(mag_diff/20));
elseif phase_diff < 0 % phase_lag controller design
    [nk,dk]=ph_lag(w_click,-phase_diff,10^(mag_diff/20));
else % constant gain design
    dk=[0 1]; % controller denominator
    nk=[0 10^(mag_diff/20)]; % controller numerator
end
%
%----- end of gr_des.m -----
%
function [num_contr,den_contr]=ph_lead(wm,phi_max,gain)
%
% Phase lead controller in transfer function form. Gain is adjusted
% linearly at w=wm
%
%  $K_{lead} = k \cdot (1+s/w_1) / (1+s/(a \cdot w_1))$ , where  $a > 1$ .
%
% Inputs:
% wm      : ang. frequency where maximum phase lead is injected
% phi_max: maximum phase in degrees
% gain    : required linear gain at wm
%
% Outputs:
% num_contr: controller numerator polynomial
% den_contr: controller denominator polynomial
%
%
if phi_max < 0 | phi_max >= 90 % check phase lead consistency
    disp('Error in ph_lead.m: 0<phi_max<90');
    return;
end
%
phi_max=phi_max*pi/180; % degrees 2 rads
alpha=(1+sin(phi_max))/(1-sin(phi_max)); % find alpha
w1=wm/sqrt(alpha); % find cut-off
%
num_contr=[1/w1 1]; % numerator of controller
den_contr=[1/(alpha*w1) 1]; % denominator of controller
controller=tf(num_contr,den_contr);

```

```

%
[mag,phase]=bode(controller,wm); % controller magnitude & phase
mag=mag(1,:);phase=phase(1,:);
%
num_contr=(gain/mag)*num_contr; % adjusting the magnitude
%
%----- end of ph_lead.m -----
%
function [num_contr,den_contr]=ph_lag(wm,phi_max,gain)
%
% Phase lag controller in transfer function form. Gain is
% adjusted - linear gain at w=wm.
%
%  $K_{lag} = k*(1+s/(a*wc))/(1+s/wc)$ , where  $a>1$  and  $s=jw$ .
%
% Inputs:
% wm      : ang. frequency where maximum phase lag is injected
% phi_max: maximum phase lag in degrees (positive!)
% gain    : required linear gain at wm
%
% Outputs:
% num_contr: controller numerator polynomial
% den_contr: controller denominator polynomial
%
%
if phi_max<0 | phi_max >= 90 % check for phase lag consistency
    disp('Error in ph_lag.m: 0<phi_max<90');
    return;
% end
%
phi_max=phi_max*pi/180; % degrees 2 rads
alpha=(1+sin(phi_max))/(1-sin(phi_max)); % find alpha
w1=wm/sqrt(alpha); % find cut-off
%
den_contr=[1/w1 1]; % denominator of controller
num_contr=[1/(alpha*w1) 1]; % numerator of controller
controller=tf(num_contr,den_contr);
%
[mag,phase]=bode(controller,wm); % controller magnitude & phase
mag=mag(1,:);phase=phase(1,:);
%
num_contr=(gain/mag)*num_contr; % scaling the magnitude
%
%----- end of ph_lag.m -----
%
function sys_controller=cseries(controller_data)
% Connect phase lead/lag controllers in series.
%
% Inputs:
% controller: 4-column matrix containing numerators and denominators of
% designed phase lead/lag controllers (one row per numerator-denominator
% pair).

```

```

%
% Outputs:
% sys_controller: overall system interconnection of controllers in series.
%
    if isempty(controller_data) % no controller designed
        sys_controller=[];
        return;
    end
%
    size_rows=size(controller_data,1); % find the no of phase lead/lag
%controllers
%
% numerator of first controller
    num_controller=[controller_data(1,1) controller_data(1,2)];
% denominator of first controller
    den_controller=[controller_data(1,3) controller_data(1,4)];
% first controller in systems form
    sys_controller=tf(num_controller,den_controller);
%
    for i=2:size_rows % start by connecting the 2nd controller
        % numerator of i-th controller
        num_temp=[controller_data(i,1) controller_data(i,2)];
        % denominator of i-th controller
        den_temp=[controller_data(i,3) controller_data(i,4)];
        sys_temp=tf(num_temp,den_temp);
        % series interconnection
        sys_controller=series(sys_controller,sys_temp);
    end
%
%----- end coseries.m -----

```

### 9.3.12 M-file to design the Pre-filter

```

function [magsys,phasesys]=fr_cl_pf(A,wc1,wc2,k,a,kd,ki,kp,w)
% function [magsys,phasesys]=fr_cl_pf(A,wc1,wc2,k,a,kd,ki,kp,w)
%
% CL Frequency response of system (with pid) and appropriate
% 2nd order filter (note that a higher order pre-filter can
% be used if necessary)
%
% INPUTS:
% A:          constant gain (usually 1 or 0dBs)
% wc1,wc2:    first and second cut-off frequencies
% k,a:        uncertain parameter vectors
% kd,ki,kp:   PID controller parameters
% w:          frequency vector
%
% OUTPUTS:
% magsys, phasesys: mag and phase of CL system (including pre-filter)
%
%
    nk=size(k,2); % size of uncertain parameter k

```

```

na=size(a,2); % size of uncertain parameter a
magsys=[]; % initialise matrix for mag.
phasesys=[]; % initialise matrix for phase
for i=1:nk
    for j=1:na
%       Numerator and denominator of plant + controller system
        num=k(i)*a(j)*[kd kp ki];
        den=[1 a(j)*(1+k(i)*kd) k(i)*a(j)*kp k(i)*a(j)*ki];
        numf=A; % numerator of pre-filter
        denf1=real(conv([1/wc1 1],[1/wc2 1])); % denominator of pre-filter
        [numov,denov]=series(num,den,numf,denf1);
        [mag,phase]=bode(numov,denov,w);
        magsys1=20*log10(mag); % translate in dBs
        phasesys1=phase;
        magsys=[magsys,magsys1];
        phasesys=[phasesys,phasesys1];
    end
end
%
%----- end of fr_cl_pf.m -----

```

## 9.4 Appendix D

Publication

**An optimisation algorithm for designing fixed-structure controllers using the QFT method**

**Keywords:** Robust Control, Quantitative Feedback Theory (QFT), Loop-shaping

### Abstract

We propose a simple optimisation algorithm for designing fixed-structure controllers for highly-uncertain plants. The method can be used to automate the loop-shaping step of the QFT design procedure and guarantees robust stability and performance to the feedback loop for all parameters in the plant's uncertainty set. To avoid over-designing the system, the algorithm can be used to minimise either the asymptotic gain or the cross-over frequency of the open-loop system. The proposed algorithm is illustrated with a simple design example, carried out within a CAD environment ("STQFT" toolbox) which is currently under development. Some preliminary results of this work have appeared in [63].

# 1. Introduction

Quantitative Feedback Theory is a systematic robust control design methodology for systems subject to large parametric or unstructured uncertainty. QFT is a graphical loop-shaping procedure, traditionally carried out on the Nichol's chart, which can be used for the control design of either SISO or MIMO uncertain systems, including the non-linear and time-varying cases [17, 28, 57]. Relative to other robust-control design methodologies, QFT offers a number of advantages, apart from its utilisation of classical control-design techniques. These include: (i) The ability to assess quantitatively the "cost of feedback" [29], (ii) the ability to take into account phase information in the design process (this is ignored in many norm-based approaches, e.g.  $\mathcal{H}_\infty$  optimal control which is based on singular values), and (iii) the ability to provide "transparency" in the design, i.e. clear tradeoff criteria between controller complexity and the feasibility of the design objectives. Note that (iii) implies in practice that QFT often results in simple controllers which are easy to implement.

The QFT design procedure is based on the two-degree of freedom feedback configuration shown in Figure 1. In this diagram  $G(p, s)$  denotes the uncertain plant, while  $K(s)$  and  $F(s)$  denote the feedback compensator and pre-filter, respectively, which are to be designed. Note that model uncertainty is described by the  $r$ -parameter vector  $p \in \mathcal{P} \subseteq \mathbb{R}^r$  taking values in the set  $\mathcal{P}$ ; it is further assumed that  $G(p, s)$  has the same number of RHP poles for all  $p \in \mathcal{P}$ . Translating the uncertainty into the frequency domain, gives rise to the plant's "uncertainty templates" which are the sets:

$$\mathcal{G}_\omega = \{G(p, j\omega) : p \in \mathcal{P}\}$$

For each fixed frequency  $\omega$ ,  $\mathcal{G}_\omega$  defines a "fuzzy region" on the Nichol's chart which describes the uncertainty of the plant at frequency  $\omega$  in terms of magnitude (in dB's) and phase (in degrees). For design purposes, we construct  $N$  uncertainty templates corresponding to a discrete set of frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$  chosen to cover adequately the system's bandwidth.

The robust performance objectives of the design include good tracking of reference input  $r(s)$  and good attenuation of the disturbance signal  $d(s)$  entering at the system's output, despite the presence of uncertainty. The robust tracking objectives are captured



by the set of inequalities:

$$\begin{aligned} \max_{p \in \mathcal{P}} \Delta \left| \frac{G(p, j\omega_i)K(j\omega_i)}{1 + G(p, j\omega_i)K(j\omega_i)} \right|_{\text{dB}} &\leq \delta(\omega_i) \\ &:= B_u(\omega_i)|_{\text{dB}} - B_l(\omega_i)|_{\text{dB}} \end{aligned}$$

for each  $i = 1, 2, \dots, N$ , i.e. if, for each frequency  $\omega_i$ , the maximum variation in closed loop gain as  $p \in \mathcal{P}$  does not exceed the maximum allowable spread in specifications  $\delta(\omega_i)$ , typically specified via two appropriate magnitude frequency responses  $B_u(\omega) = |B_u(j\omega)|$  and  $B_l(\omega) = |B_l(j\omega)|$ . Note that it is not necessary to bound the actual gain (but only the gain spread) since we assume that, (i) no uncertainty is associated with the feedback controller  $K(s)$ , and (ii) the pre-filter  $F(s)$  can provide arbitrary scaling to the closed-loop gain.

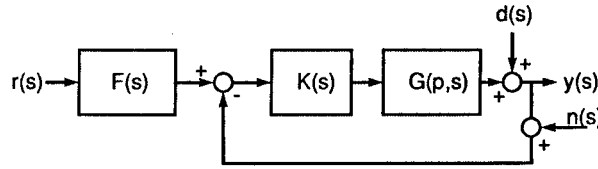


Figure 9.1: Feedback Configuration

The robust disturbance -rejection objective can be satisfied by bounding the sensitivity function, i.e. by imposing constraints of the form

$$\max_{p \in \mathcal{P}} \left| \frac{1}{1 + G(p, j\omega_i)K(j\omega_i)} \right| \leq D(\omega_i)$$

for a (subset) of the design frequencies  $\{\omega_1, \omega_2, \dots, \omega_N\}$ . Again these are typically specified via an appropriate magnitude frequency-response  $D(\omega) = |D(j\omega)|$ .

Robust stability is enforced by ensuring that: (i) no unstable pole-zero cancellations occur between the plant and the controller (for every  $p \in \mathcal{P}$ ), (ii) the nominal open-loop frequency response  $L_o(j\omega) = G(p_o, j\omega)K(j\omega)$  (defined for any  $p_o \in \mathcal{P}$ ) does not cross the  $-1$  point (i.e. the  $(-180^\circ, 0)$  point on Nichol's chart) and makes a total number of (anti-clockwise) encirclements around it equal to the number of unstable poles of  $L_o(s) = G(p_o, s)K(s)$ , and (iii) That no (perturbed) open-loop response crosses the  $-1$  point, i.e.

$$-1 \notin \bigcup_{\omega \in \mathbb{R}} K(j\omega)\mathcal{G}_\omega$$

Note that condition (i) is automatically satisfied if  $K(s)$  is restricted to be stable and minimum-phase, while conditions (ii) and (iii) can be easily tested graphically [13, 12]. In practice, a more severe condition than (iii) is imposed: To establish a minimum amount of damping, it is required that the nominal open-loop frequency response does not penetrate a closed contour in the Nichol's chart ( $\mathcal{U}$ -contour); this is constructed from an appropriate  $M$ -circle and information about high-frequency uncertainty of the plant [17, 28].

The robust tracking and disturbance rejection objectives have been formulated as gain inequalities of the closed-loop transfer functions (sensitivity and complementary sensitivity) at the design frequencies. For the purposes of QFT design, these inequalities must be translated into constraints on the nominal open-loop response  $L_o(j\omega)$ . This procedure results into a number of contours ("Horowitz templates"  $f_i^t(\phi)$  and "disturbance-rejection templates"  $f_i^d(\phi)$ ) for each frequency  $\omega_i$ ,  $i = 1, 2, \dots, N$ ; these are functions of the phase variable  $\phi \in (-360^\circ, 0^\circ]$ . Thus, robust tracking is satisfied at frequency  $\omega_i$  if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$  where  $\arg L_o(j\omega_i) = \phi_i$ ; similarly, robust disturbance-rejection is attained at frequency  $\omega_i$  if  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi)$ . The robust-performance templates (Horowitz and disturbance-rejection) can be easily constructed (within an arbitrary gain tolerance and for a discretised phase-grid) using a simple bisection algorithm.

In conclusion, assuming that the condition prohibiting unstable pole/zero cancellations between the plant and the controller is independently verified, the following conditions guarantee robust-stability and performance:

- The winding number of the nominal open-loop system  $L_o(j\omega)$  around the  $-1$  point is equal to the number of RHP poles of  $L_o(s)$ .
- The nominal open-loop frequency response  $L_o(j\omega)$  does not penetrate the  $\mathcal{U}$  contour.
- The following inequalities are satisfied for all  $i = 1, 2, \dots, n$ :

$$|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$$

and

$$|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi_i)$$

in which  $\phi_i = \arg L_o(j\omega_i)$ ; these inequalities correspond to the robust tracking and robust disturbance-rejection specifications respectively.

## 2. Formulation of QFT constraints

In this section we first formulate the QFT robust stability and performance constraints as a feasibility programme. This leads to an optimisation algorithm for carrying out optimal QFT designs using a family of simple fixed-structure compensators. This is in contrast to other approaches (e.g. [6]) which optimise the frequency response of the open-loop system.

As was shown in the last section, the QFT robust-stability and performance objectives can be translated to graphical constraints on the Nichols chart. The constraints associated with robust-performance (“Horowitz” and “disturbance-rejection” templates) correspond to open contours, i.e. they split the Nichols chart in two regions (for each design frequency), the high and low-gain region. To meet the tracking or disturbance-rejection objective, the nominal open-loop frequency response point  $L_o(j\omega_i)$  must be placed on the high-gain region of the contour, i.e. forced to satisfy the inequality  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^t(\phi_i)$  (tracking) or  $|L_o(j\omega_i)|_{\text{dB}} \geq f_i^d(\phi_i)$  (disturbance rejection), where  $\arg L_o(j\omega_i) = \phi_i$ . In contrast, the robust-stability template ( $\mathcal{U}$ -contour) is a closed contour containing the critical point. To construct the  $\mathcal{U}$ -contour, we start from the definition of an  $M$ -circle ( $M > 1$ ) in the Nyquist plane ( $u, v$ ):

$$\left(u + \frac{M^2}{M^2 - 1}\right)^2 + v^2 = \frac{M^2}{(M^2 - 1)^2}$$

which is a circle of centre  $(u, v) = \left(-\frac{M^2}{M^2 - 1}, 0\right)$  and radius  $R = \frac{M}{M^2 - 1}$ .

Since in this case ( $M > 1$ ) the  $M$ -circle does not contain the origin, it is clear that in the Nichol’s chart it is defined only for an interval of phases, and is symmetric around the phase line  $\phi = -180^\circ$ .

In fact drawing the tangents to the circle from the origin, it is clear that (see Figure 2)

$$\psi_{\max} = \sin^{-1} \left( \frac{1}{M} \right)$$

and hence the  $M$ -circle is defined on the Nichol’s chart only for the phase interval

$$-180^\circ - \sin^{-1} \left( \frac{1}{M} \right) \leq \phi \leq -180^\circ + \sin^{-1} \left( \frac{1}{M} \right)$$

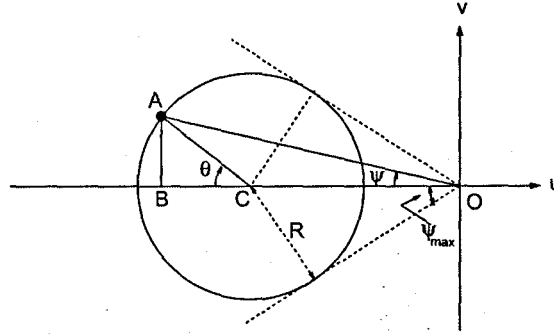


Figure 9.2:  $M$ -circle in Nyquist plane ( $M > 1$ )

Next we need to derive the equation of the  $M$ -circle in terms of magnitude ( $m$ ) and phase ( $\phi$ ), where

$$m = \sqrt{u^2 + v^2} \quad \text{and} \quad \phi = \arctan \frac{v}{u}$$

Referring to Figure 2, it follows by simple geometry that

$$\begin{aligned} m \cos \psi &= \frac{M^2}{M^2 - 1} + \frac{M}{M^2 - 1} \cos \theta \\ m \sin \psi &= \frac{M}{M^2 - 1} \sin \theta \end{aligned}$$

Eliminating variable  $\theta$  using the trigonometric identity  $\sin^2 \theta + \cos^2 \theta = 1$  results in the second-order equation:

$$m^2 - \frac{2M^2 \cos \psi}{M^2 - 1} m + \frac{M^2}{M^2 - 1} = 0$$

which can be solved as:

$$m = \frac{M^2}{M^2 - 1} \left( \cos \psi \pm \sqrt{\frac{1}{M^2} - \sin^2 \psi} \right)$$

Thus, using the substitution  $\phi = -180^\circ - \psi$ , the  $M$  circle in the Nichols chart is a closed contour which may be decomposed into the union of the graphs of the two functions,

$$\begin{aligned} M^+(\phi) &= 20 \log_{10} \left( -\cos \phi + \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right) \\ &\quad + 20 \log_{10} \left( \frac{M^2}{M^2 - 1} \right) \end{aligned}$$

and

$$\begin{aligned} M^-(\phi) &= 20 \log_{10} \left( -\cos \phi - \sqrt{\frac{1}{M^2} - \sin^2 \phi} \right) \\ &\quad + 20 \log_{10} \left( \frac{M^2}{M^2 - 1} \right) \end{aligned}$$

Following Horowitz [29], the  $\mathcal{U}$  contour is obtained by translating  $M^-(\phi)$  vertically by  $V_\infty$  dB's, where

$$V_\infty = \lim_{\omega \rightarrow \infty} \left( \max_{p \in \mathcal{P}} |G(p, j\omega)|_{\text{dB}} - |G(p_o, j\omega)|_{\text{dB}} \right)$$

This gives the  $\mathcal{U}$  contour as the union of the graphs of the two functions:

$$\mathcal{U}^+(\phi) = M^+(\phi) \quad \text{and} \quad \mathcal{U}^-(\phi) = M^-(\phi) - V_\infty$$

over the phase interval

$$\begin{aligned} \phi_l &:= -180^\circ - \sin^{-1} \left( \frac{1}{M} \right) \leq \phi \\ &\leq -180^\circ + \sin^{-1} \left( \frac{1}{M} \right) := \phi_h \end{aligned}$$

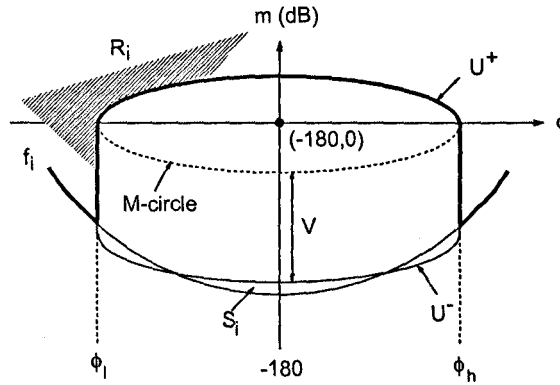


Figure 9.3: M-circle and  $\mathcal{U}$ -contour in Nichols chart ( $M > 1$ )

The ultimate objective of this section is to characterise the regions of the Nichols chart in which the open-loop frequency response point  $L_o(j\omega_i)$  can lie in order to satisfy the robust stability and performance constraints. To this purpose define the composite function

$$f_i^m(\phi) = \max\{f_i^t(\phi), f_i^d(\phi)\}$$

where the maximum is taken pointwise in  $\phi \in (-360^\circ, 0^\circ]$ . Further define,

$$\begin{aligned} f_i(\phi) &= f_i^m(\phi) && \text{for } \phi \leq \phi_l, \\ &= \max\{f_i^m(\phi), \mathcal{U}^+(\phi)\} && \text{for } \phi_l < \phi < \phi_h \\ &= f_i^m(\phi) && \text{for } \phi \geq \phi_h \end{aligned}$$

Also let  $\Phi_i = \{\phi : \phi_l < \phi < \phi_h, f_i^m(\phi) \leq \mathcal{U}^-(\phi)\}$ . Then, the robust stability and performance constraints at frequency  $\omega_i$  are satisfied iff  $L_o(j\omega_i) \in R_i \cup S_i$ , where

$$R_i = \{L_o(j\omega_i) : |L_o(j\omega_i)|_{\text{dB}} \geq f_i(\phi), \phi = \arg L_o(j\omega_i)\}$$

and

$$S_i = \{L_o(j\omega_i) : f_i(\phi) \leq |L_o(j\omega_i)|_{\text{dB}} \leq \mathcal{U}^-(\phi), \\ \phi = \arg L_o(j\omega_i) \in \Phi_i\}$$

An illustration of the region  $R_i \cup S_i$  is given in Figure 3. Note that in practice, when a performance constraint is active, we typically have  $\Phi_i = S_i = \emptyset$ . This is because performance objectives are normally specified at low frequencies, rarely exceeding the closed-loop bandwidth of the system. However, our formulation allows us to take into account “unconstrained” design frequencies, i.e. frequencies at which no performance inequalities apply. For such a frequency  $\omega_i$  we would have  $f_i^m(\phi) = -\infty$  and hence  $\Phi_i = (\phi_l, \phi_h)$ ,  $S_i = \{L_o(j\omega_i) : |L_o(j\omega_i)|_{\text{dB}} \leq \mathcal{U}^-(\phi), \phi = \arg L_o(j\omega_i) \in (\phi_l, \phi_h)\}$  (i.e. the region below the  $\mathcal{U}$  contour), while  $R_i \cup S_i$  would represent the region outside the  $\mathcal{U}$ -contour.

The conditions that guarantee the robust-stability and robust-performance specifications at the design frequencies can now be summarised by the following two graphical tests:

- The winding number of the nominal open-loop system  $L_o(j\omega)$  around the  $-1$  point is equal to the number of RHP poles of  $L_o(s)$ .
- For each frequency  $\omega_i$ ,  $L_o(j\omega_i) \in R_i \cup S_i$ .

Again, it is assumed that no unstable pole/zero cancellations occur between the controller and  $G(p, s)$  for every  $p \in \mathcal{P}$ , a condition which must be checked independently. Of course, similarly to any QFT-based method, these tests do not really guarantee that  $L_o(j\omega)$  does not enter the  $\mathcal{U}$  contour (at frequencies other than the design frequencies). This, however, does not cause a problem in practice, provided a reasonably large set of design frequencies is selected near cross-over, or, alternatively, by slightly tightening the specifications by means of an appropriate tolerance.

### 3. Optimisation algorithm

In this section we outline an optimisation algorithm for designing fixed-structure compensators of certain types subject to the QFT constraints developed earlier. Every design (i.e. loop shaping of  $L_o(j\omega)$ ) which satisfies the two graphical tests of the last section is in principle “admissible”, i.e. satisfies the robust stability and the robust-performance objectives. Since in general many different designs may be admissible, we require a method of classifying them by formulating an appropriate optimisation criterion. Adopting the arguments of Horowitz and Sidi [29, 30], such a criterion must penalise the “over-design” of the system, e.g. an unnecessarily high closed-loop bandwidth, since this increases the “cost of feedback” in terms of sensor-noise amplification and potential instability due to high-frequency unmodelled dynamics/parasitics. Appropriate “cost functions” to be minimised include the following quantities:

- Open-loop cross-over frequency (nominal or worst-case).
- Closed-loop bandwidth (nominal or worst-case).
- Asymptotic open-loop gain.

Each of the above measures can be calculated in a straightforward manner from the frequency response of the system. Note that all these criteria depend monotonically on the system’s open-loop gain.

The types of compensators considered in this paper are listed below. Note that some of these must be used under appropriate relative-degree assumptions satisfied by the transfer function of the plant.

1. PID:  $K(s) = k_p + k_d s + \frac{k_i}{s}$
2. First-order lead/lag:  $K(s) = \frac{k(s+b)}{s+a}$
3. Second-order with complex-poles (zeros):  $K(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$  (or  $s^2 + 2\zeta\omega_n s + \omega_n^2$ ).

PID and phase lead/lag controllers are widely used in industry because they are simple and easy to tune. Thus optimal controllers of the first two types may provide simple solutions to robust control designs based on the QFT method. Note also that every rational controller of arbitrary complexity can be constructed from cascade

interconnections of controllers in (2) and (3) above. Thus, it is possible to improve the design continuously by building higher-order controllers in a step-by-step procedure: At each step the optimisation algorithm is carried out (for one of the three controller structures) and the resulting optimal controller  $K(s)$  is accumulated into the nominal open-loop system by redefining  $L_o(s) \leftarrow L_o(s)K(s)$ . This process may continue until a satisfactory design is obtained, or until the cost fails to decrease significantly. Of course, the controller resulting from this procedure will not, in general, be optimal over the higher-order controller set!

The proposed algorithm is based on the fact that fixing the phase of the compensator at two distinct frequencies determines the compensator uniquely *up to scaling*. Thus, the phase-response of the nominal open-loop system is also completely determined, and only a simple calculation is needed to determine the minimum amount of gain required to meet the QFT robust stability and performance specifications (if these are feasible). Geometrically, this corresponds to shifting the frequency response of  $L_o(s)$  vertically in the Nichol's chart by the minimum gain required to place the the points  $L_o(j\omega_i)$  in the  $R_i \cup S_i$  regions while simultaneously satisfying the Nyquist stability encirclement condition. Repeating this procedure for all possible phase combinations (suitably discretised) will eventually produce the optimal design (if one exists). Next, we consider each controller type in turn:

### 3.1 PID controller:

*Theorem 1: [63]* (i) Let  $K(s) = k_p + k_d s + \frac{k_i}{s}$  with  $k_p, k_d, k_i$  real parameters. Suppose that  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  where  $\omega_i \neq \omega_j$ . Then the matrix:

$$A_{ij} = \begin{pmatrix} 1 & -\frac{1}{\omega_i^2} & -\frac{\tan(\psi_i)}{\omega_i} \\ 1 & -\frac{1}{\omega_j^2} & -\frac{\tan(\psi_j)}{\omega_j} \end{pmatrix}$$

has full (row) rank. Let  $(V^{ij}) = [V_1^{ij} \ V_2^{ij} \ V_3^{ij}]' \in \mathbb{R}^3$  be a (real) non-zero vector in the (one-dimensional) kernel of  $A_{ij}$ . Then,

$$\begin{pmatrix} k_d \\ k_i \\ k_p \end{pmatrix} = \lambda \begin{pmatrix} V_1^{ij} \\ V_2^{ij} \\ V_3^{ij} \end{pmatrix}$$



where  $\lambda$  is an arbitrary real constant. Moreover, the gain and phase of the controller at any frequency  $\omega$  is given by:

$$|K(j\omega)| = |\lambda| \sqrt{(V_3^{ij})^2 + \left(V_1^{ij}\omega - \frac{V_2^{ij}}{\omega}\right)^2} := |\lambda| C_{ij}(\omega)$$

and

$$\arg K(j\omega) := \psi(\omega) = \arctan \left( \frac{V_1^{ij}\omega - \frac{V_2^{ij}}{\omega}}{V_3^{ij}} \right)$$

respectively.

(ii) If the controller gains  $k_p$ ,  $k_i$  and  $k_d$  are restricted to be non-negative, then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  are feasible if no two of the three scalars  $V_1^{ij}$ ,  $V_2^{ij}$  and  $V_3^{ij}$  have opposite signs. In this case,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{V_1^{ij}\omega_k - \frac{V_2^{ij}}{\omega_k}}{V_3^{ij}} \right)$$

Also,  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $|\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \notin [\phi_l \phi_h]$ .
- $|\lambda| \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)} \geq |\lambda| \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$   
or  $|\lambda| \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C_{ij}(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C_{ij}(\omega_k)$  is defined in part (i).

*Proof:* (i) The frequency response of the PID controller is given as

$$K(j\omega) = k_p + jk_d\omega - j\frac{k_i}{\omega}$$

with gain and phase

$$|K(j\omega)| = \sqrt{k_p^2 + \left(k_d\omega - \frac{k_i}{\omega}\right)^2}$$

and

$$\arg K(j\omega) = \arctan \left( \frac{k_d\omega - \frac{k_i}{\omega}}{k_p} \right)$$

respectively. Now suppose we fix  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two frequencies  $\omega_i \neq \omega_j$ . Then,

$$\begin{aligned} k_d - \frac{k_i}{\omega_i^2} - \frac{k_p \tan(\psi_i)}{\omega_i} &= 0 \\ k_d - \frac{k_i}{\omega_j^2} - \frac{k_p \tan(\psi_j)}{\omega_j} &= 0 \end{aligned}$$

which can be written in matrix form as:

$$\begin{pmatrix} 1 & -\frac{1}{\omega_i^2} & -\frac{\tan(\psi_i)}{\omega_i} \\ 1 & -\frac{1}{\omega_j^2} & -\frac{\tan(\psi_j)}{\omega_j} \end{pmatrix} \begin{pmatrix} k_d \\ k_i \\ k_p \end{pmatrix} = 0$$

Clearly,  $\text{Rank}(A_{ij}) = 2$ , since  $\omega_i \neq \omega_j$  and thus the controller parameter vector is constrained to lie in the one-dimensional subspace  $\text{Ker}(A_{ij})$ . Writing  $\text{Ker}(A_{ij}) = \lambda[V_1^{ij} \ V_2^{ij} \ V_3^{ij}]'$  gives the required expressions for  $k_d$ ,  $k_i$  and  $k_p$  from which the magnitude and phase expressions of  $K(j\omega)$  follow immediately. (ii) It is clear that when the controller gains are restricted to be non-negative, the three scalars  $V_1^{ij}$ ,  $V_2^{ij}$  and  $V_3^{ij}$  must be either non-positive or non-negative simultaneously. The conditions for  $L_o(j\omega_k) \in R_k \cup S_k$  then follow immediately from the formulation of the QFT constraints given in the previous section.  $\square$

Theorem 1 shows that fixing the phase of the PID controller between  $-90^\circ$  and  $90^\circ$  at two distinct frequencies, fixes the phase of the controller at every frequency. The Nyquist plot of the PID controller (a vertical straight line with real part  $k_p$ ) shows geometrically that in this case the three controller gains are uniquely determined (up to scaling) provided that  $90^\circ < \psi_i < \psi_j < 90^\circ$  for  $\omega_i < \omega_j$ .

The one-dimensional kernel of matrix  $A_{ij}$  can be calculated efficiently using the singular value decomposition algorithm. Alternatively, the three gains can be calculated analytically in terms of the parameters  $\omega_i$ ,  $\omega_j$ ,  $\psi_i$  and  $\psi_j$ ; this, however, does not seem to result in any significant computational advantage.

If the pure-derivative term in the controller is considered to be undesirable, the controller can be modified to the form:

$$K'(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1 + s\tau}$$

where  $\tau$  is a (fixed) sufficiently small parameter. In this case, Theorem 1 can be applied with minor modifications by redefining the uncertain plant as

$$G'(p, s) = \frac{G(p, s)}{1 + s\tau}$$

and solving for the new variables  $k_p' = k_p + k_i\tau$ ,  $k_i' = k_i$  and  $k_d' = k_d + k_p\tau$ . See [63] for details.

### 3.2 Phase lead/lag controller

*Theorem 2:* Let  $K(s) = k(s + b)/(s + a)$  with  $a > b > 0$  ("phase lead" controller).

Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $0 < \psi_i < 90^\circ$  and  $0 < \psi_j < 90^\circ$  are feasible if and only if the following two conditions are satisfied:

$$\lambda := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} > 0$$

and

$$c := \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

In this case, the quadratic equation:

$$b^2 + \lambda b + c = 0$$

has one positive root  $b_+$  and one negative root  $b_-$  and the controller parameters  $b$  and  $a$  are determined uniquely as  $b = b_+$  and  $a = b_+ + \lambda = -b_-$ . In addition,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan\left(\frac{\omega_k}{b}\right) - \arctan\left(\frac{\omega_k}{a}\right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l, \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$   
or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = \sqrt{(b^2 + \omega^2)/(a^2 + \omega^2)}$ .

*Proof:* The frequency response of the phase-lead controller is given as

$$K(j\omega) = k \frac{j\omega + b}{j\omega + a}$$

with gain and phase

$$|K(j\omega)| = k \sqrt{\frac{\omega^2 + b^2}{\omega^2 + a^2}}$$

and

$$\arg K(j\omega) = \arctan\left(\frac{\omega}{b}\right) - \arctan\left(\frac{\omega}{a}\right)$$

respectively. Now suppose we fix  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two frequencies  $\omega_i \neq \omega_j$ . Then,

$$\begin{aligned} \arctan\left(\frac{\omega_i}{b}\right) - \arctan\left(\frac{\omega_i}{a}\right) &= \psi_i \\ \arctan\left(\frac{\omega_j}{b}\right) - \arctan\left(\frac{\omega_j}{a}\right) &= \psi_j \end{aligned}$$

Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $0 < \psi_i < 90^\circ$  and  $0 < \psi_j < 90^\circ$  are feasible if and only if the following two conditions are satisfied:

$$\lambda := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} > 0$$

and

$$c := \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

In this case, the quadratic equation:

$$b^2 + \lambda b + c = 0$$

has one positive root  $b_+$  and one negative root  $b_-$  and the controller parameters  $b$  and  $a$  are determined uniquely as  $b = b_+$  and  $a = b_+ + \lambda = -b_-$ . In addition,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan\left(\frac{\omega_k}{b}\right) - \arctan\left(\frac{\omega_k}{a}\right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l, \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$   
or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{\text{dB}})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = \sqrt{(b^2 + \omega^2)/(a^2 + \omega^2)}$ .

*Proof:* The frequency response of the phase-lead controller is given as

$$K(j\omega) = k \frac{j\omega + b}{j\omega + a}$$

with gain and phase

$$|K(j\omega)| = k \sqrt{\frac{\omega^2 + b^2}{\omega^2 + a^2}}$$

and

$$\arg K(j\omega) = \arctan\left(\frac{\omega}{b}\right) - \arctan\left(\frac{\omega}{a}\right)$$

respectively. Now suppose we fix  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two frequencies  $\omega_i \neq \omega_j$ . Then,

$$\begin{aligned} \arctan\left(\frac{\omega_i}{b}\right) - \arctan\left(\frac{\omega_i}{a}\right) &= \psi_i \\ \arctan\left(\frac{\omega_j}{b}\right) - \arctan\left(\frac{\omega_j}{a}\right) &= \psi_j \end{aligned}$$

Using the trigonometric identity:

$$\tan(\alpha - \beta) = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta}$$

we get (after some algebra),

$$\omega_i^2 \tan \psi_i - \omega_i a + \omega_i b + ab \tan \psi_i = 0 \quad (9.14)$$

$$\omega_j^2 \tan \psi_j - \omega_j a + \omega_j b + ab \tan \psi_j = 0 \quad (9.15)$$

Multiplying the first equation by  $\tan \psi_j$ , the second equation by  $\tan \psi_i$  and subtracting the resulting two equations gives

$$a = b + \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} := b + \lambda$$

Since  $a > b$  for a phase-lead controller, we must have  $\lambda > 0$ . Substituting for  $a = b + \lambda$  in (1) leads to the quadratic equation:

$$b^2 + \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} b + \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} = 0$$

This must have a positive root  $b_+$  if the constraints are feasible, so that  $a = b_+ + \lambda > b_+ = b > 0$ . To see that *at most* one of the two roots of the quadratic equation

$$b = \frac{-\lambda \pm \sqrt{\lambda^2 - 4c}}{2}$$

is positive, note that the transfer functions  $(s + b)/(s + a)$  and  $(s - a)/(s - b)$  have identical phase responses; hence if one root of the quadratic is  $b$ , the other root must be  $-a$ . Formally, when  $\lambda > 0$  the roots of the quadratic can be classified as follows:

- $c < 0$ : One positive ( $b_+$ ) and one negative ( $b_-$ ) root.
- $c = 0$ : Zero and negative ( $b = -\lambda$ ) roots.
- $0 < c \leq \lambda^2/4$ : Here  $\sqrt{\lambda^2 - 4c} < \lambda$  so both roots are negative.
- $c > \lambda^2/4$ : Complex conjugate roots.

and so parameters  $a$  and  $b$  with  $a > b > 0$  are uniquely determined from the two phase conditions when  $\lambda > 0$  and  $c < 0$ . To show that  $b_- = -a$  note that

$$a = b_+ + \lambda = \frac{-\lambda + \sqrt{\lambda^2 - 4c}}{2} + \lambda = \frac{\lambda + \sqrt{\lambda^2 - 4c}}{2} = -b_-$$

The phase equation for  $L_o(j\omega_k)$  is immediate, while the gain inequalities on  $k$  for  $L_o(j\omega_k) \in R_k \cup S_k$  follow directly from the discussion of the previous section on noting that  $|K(j\omega_k)| = kC(\omega_k)$ .  $\square$

**Example:** Consider the following cases:

- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = 10^\circ$  and  $\psi_j = 30^\circ$ : We obtain  $\lambda = 11.9339$ ,  $c = -66.6806$  and so the constraints are feasible. The quadratic equation gives  $b = b_+ = 4.1467$  and  $a = b_+ + \lambda = -b_- = 16.0806$ .
- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = 60^\circ$  and  $\psi_j = 10^\circ$ : We obtain  $\lambda = 0.6785$ ,  $c = 0.6083$  and the roots of the quadratic are complex:  $b_{1,2} = -0.3393 \pm j0.7023$ . The constraints are infeasible.
- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = -10^\circ$  and  $\psi_j = 30^\circ$ : Clearly the constraints are infeasible for a lead (or lag) controller. We obtain  $\lambda = 1.1905$ ,  $c = 7.7518$  and the quadratic has complex roots  $b_{1,2} = -0.5953 \pm j2.7198$ .
- $\omega_i = 1$  rads/s,  $\omega_j = 4$  rads/s,  $\psi_i = -10^\circ$  and  $\psi_j = -30^\circ$ : Clearly constraints are infeasible for phase-lag controller (but not for a phase-lead controller). We obtain  $\lambda = -11.9339$ ,  $c = -66.6806$  while the quadratic equation gives  $b_+ = 16.0806$  and  $b_- = -4.1467$ .

The corresponding result for a phase-lag controller is as follows:

*Theorem 2'*: Let  $K(s) = k(s + b)/(s + a)$  with  $b > a > 0$  ("phase lag" controller).

Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $-90^\circ < \psi_i < 0^\circ$  and  $-90^\circ < \psi_j < 0^\circ$  are feasible if and only if the following two conditions are satisfied:

$$\lambda := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_i \tan \psi_j}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

and

$$c := \frac{\omega_i \omega_j (\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} < 0$$

In this case, the quadratic equation:

$$b^2 + \lambda b + c = 0$$

has one positive root  $b_+$  and one negative root  $b_-$  and the controller parameters  $b$  and  $a$  are determined uniquely as  $b = b_+$  and  $a = b_+ + \lambda = -b_-$ . In addition,

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan\left(\frac{\omega_k}{b}\right) - \arctan\left(\frac{\omega_k}{a}\right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l, \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$   
or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l, \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = \sqrt{(b^2 + \omega^2)/(a^2 + \omega^2)}$ .

*Proof:* Along similar lines to the proof of Theorem 2.  $\square$

Theorems 2 and 2' show that fixing the phases of the phase lead or phase lag controller in the intervals  $(0^\circ, 90^\circ)$  or  $(-90^\circ, 0^\circ)$  respectively determines uniquely the dynamic part of the controller when the constraints are feasible. Feasibility of the constraints is easily checked from two sign conditions, and the controller parameters are determined by solving a quadratic equation.

### 3.3 Second-order controller with complex poles or zeros

*Theorem 3:* (i) Let  $K(s) = k/(s^2 + 2\zeta\omega_n s + \omega_n^2)$  with  $\omega_n > 0$  and  $0 < \zeta < 1$  ("complex-pole second-order lag"). Then the constraints  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  for two distinct frequencies  $\omega_i \neq \omega_j$  with  $-180^\circ < \psi_i < 0^\circ$  and  $-180^\circ < \psi_j < 0^\circ$  and  $\psi_i \neq \psi_j$  are feasible if and only if the following conditions are satisfied:

- If  $\psi_i \neq -90^\circ$  and  $\psi_j \neq -90^\circ$ ,

$$\lambda = \frac{\omega_j \tan \psi_j - \omega_i \tan \psi_i}{\omega_i \tan \psi_j - \omega_j \tan \psi_i} > 0$$

and

$$0 < \zeta := \frac{\tan \psi_i}{2} \left( \sqrt{\frac{\omega_j \lambda}{\omega_i}} - \sqrt{\frac{\omega_i}{\lambda \omega_j}} \right) < 1 \quad (9.16)$$

in which case  $\omega_n$  and  $\zeta$  are defined uniquely as  $\omega_n = \sqrt{\omega_i \omega_j \lambda}$  and via (3) respectively.

- If  $\psi_j = -90^\circ$  then either of the two conditions must hold: (i)  $\psi_i \in (-90^\circ, 0^\circ)$  and  $\omega_j < \omega_i$ , or (ii)  $\psi_i \in (-180^\circ, -90^\circ)$  and  $\omega_j > \omega_i$ , in addition to the condition

$$\zeta := \frac{(\omega_j^2 - \omega_i^2) \tan \psi_i}{2\omega_i \omega_j} < 1 \quad (9.17)$$

in which case  $\omega_n$  and  $\zeta$  are uniquely determined as  $\omega_n = \omega_j$  and via (4), respectively.

- If  $\psi_i = -90^\circ$  then either of the two conditions must hold: (i)  $\psi_j \in (-90^\circ, 0^\circ)$  and  $\omega_i < \omega_j$ , or (ii)  $\psi_j \in (-180^\circ, -90^\circ)$  and  $\omega_i > \omega_j$ , in addition to the condition

$$\zeta := \frac{(\omega_i^2 - \omega_j^2) \tan \psi_j}{2\omega_i \omega_j} < 1 \quad (9.18)$$

in which case  $\omega_n$  and  $\zeta$  are uniquely determined as  $\omega_n = \omega_i$  and via (5), respectively.

(ii) When the phase conditions are feasible we have

$$\arg L_o(j\omega_k) := \phi_k = \arg G_o(j\omega_k) + \arctan \left( \frac{2\zeta\omega_n\omega_k}{\omega_n^2 - \omega_k^2} \right)$$

and  $L_o(j\omega_k) \in R_k \cup S_k$  iff

- $k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \notin [\phi_l \phi_h]$ .
- $k \geq \frac{10^{(f_k(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) \geq M^-(\phi_k)$ .
- $\frac{10^{(M^-(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)} \geq k \geq \frac{10^{(f_k^m(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  or  $k \geq \frac{10^{(M^+(\phi_k) - |G_o(j\omega_k)|_{dB})/20}}{C(\omega_k)}$  when  $\phi_k \in [\phi_l \phi_h]$  and  $f_k^m(\phi_k) < M^-(\phi_k)$ .

where  $C(\omega) = 1/\sqrt{(\omega_n - \omega)^2 + 4\zeta^2\omega_n^2\omega^2}$ .

*Proof:* The frequency response of the controller is given by

$$K(j\omega) = \frac{k}{\omega_n^2 - \omega^2 + 2j\zeta\omega_n\omega}$$

from which its magnitude and phase responses can be obtained as:

$$|K(j\omega)| = \frac{k}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} = kC(\omega)$$

and

$$\arg K(j\omega) = \arctan \left( \frac{2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \right)$$



respectively. Setting  $\arg K(j\omega_i) = \psi_i$  and  $\arg K(j\omega_j) = \psi_j$  gives

$$\frac{2\zeta\omega_n\omega_i}{\omega_n^2 - \omega_i^2} = \tan \psi_i \quad \text{and} \quad \frac{2\zeta\omega_n\omega_j}{\omega_n^2 - \omega_j^2} = \tan \psi_j \quad (9.19)$$

for  $\psi_i \neq -90^\circ$  and  $\psi_j \neq -90^\circ$ . Solving simultaneously the above two equations gives:

$$\omega_n^2 = \frac{\omega_i\omega_j(\omega_j \tan \psi_j - \omega_i \tan \psi_i)}{\omega_i \tan \psi_j - \omega_j \tan \psi_i}$$

which defines  $\omega_n$  uniquely iff  $\lambda > 0$ . Substituting into (6) then gives the expression for  $\zeta$  and the corresponding condition for an under-damped response ( $0 < \zeta < 1$ ). When  $\psi_j = -90^\circ$  we must have  $\omega_n = \omega_j$  and hence  $\zeta$  is given by (5). This is positive when  $(\omega_j - \omega_i) \tan \psi_i > 0$  from which the two stated conditions follow. Finally, the phase equation for  $L_o(j\omega_k)$  follows immediately, while the gain conditions for  $L_o(j\omega_k) \in R_k \cup S_k$  can be derived from the discussion in the previous section on QFT constraints.  $\square$

Again, Theorem 3 shows that fixing the phase of the controller at two distinct frequencies, determines completely the dynamic part of the controller, when the constraints are feasible. In the Theorem formulation the controller is restricted to be under-damped. This restriction can be removed, if required, by ignoring throughout the  $\zeta < 1$  condition. An almost identical procedure may be used to determine the dynamic part of a minimum-phase non-proper controller  $k(s^2 + 2\zeta\omega_n s + \omega_n^2)$  from its two phases in the interval  $(0^\circ, 180^\circ)$  at two distinct frequencies  $\omega_i$  and  $\omega_j$ ; details are omitted.

In all three cases considered above we have derived simple gain conditions on the nominal open-loop gain, so that the QFT robust stability and performance constraints are satisfied. These are of the form  $L_o(j\omega_k) \in R_k \cup S_k$ , which for a fixed phase  $\arg L_o(j\omega_k) = \phi_k$ , correspond to gain intervals

$$[k_1(\omega_i, \phi_k), k_2(\omega_i, \phi_k)] \cup [k_3(\omega_i, \phi_k), \infty]$$

where  $i = 1, 2, \dots, N$ . Thus the optimisation problem has the form: "Minimise the optimisation criterion (i.e. cross-over frequency, closed-loop bandwidth, asymptotic open-loop gain, etc), so that for each design frequency  $\omega_i$ ,  $i = 1, 2, \dots, N$ ,  $L_o(j\omega_k) \in S_k \cup R_k$  and Nyquist's encirclement criterion is satisfied". Since for the three types of controllers described above the phase of the nominal open-loop system is completely determined once two controller phases have been fixed, we can use the following algorithm for solving the optimisation problem:

## Optimisation algorithm:

1. Obtain a phase array  $\Phi$  by discretising the phase interval  $(-360^\circ \ 0^\circ]$ .
2. Select any two distinct frequencies  $\omega_i$  and  $\omega_j$  in the set of design frequencies  $(\omega_1, \omega_2, \dots, \omega_N)$ .
3. Calculate the phase intervals  $\Phi_k \subseteq \Phi$  and  $\Phi_l \subseteq \Phi$  in which the nominal open loop phase  $\arg L_o(j\omega)$  can vary at  $\omega = \omega_k$  and  $\omega = \omega_l$  respectively. These depend on the type of controller to be designed, e.g. for a PID controller they lie within  $\pm 90^\circ$  of  $\arg G(p_o, \omega_k)$  and  $\arg G(p_o, \omega_l)$ , etc.
4. Initialise an  $m \times n$  array  $F$  where  $m$  and  $n$  are the sizes of  $\Phi_k$  and  $\Phi_l$  respectively, to contain the value of the objective function (cross-over frequency, closed-loop bandwidth, asymptotic gain etc.) for each phase pair. Also, initialise three  $m \times n$  controller parameter arrays to contain the parameters  $(k_p, k_d, k_i)$  for a PID controller,  $(k_o, a, b)$  for a phase-lead/lag controller, or  $(k_o, \omega_n, \zeta)$  for a second-order controller with complex poles/zeros.
5. For each  $(\Phi_k(i), \Phi_l(j)) \in \Phi_k \times \Phi_l \subseteq \mathbb{R}^{m \times n}$ :
  - (a) Calculate  $\psi_i = \Phi_k(i) - \arg G(p_o, \omega_k)$  and  $\psi_j = \Phi_l(j) - \arg G(p_o, \omega_l)$ .
  - (b) Determine a controller  $K(s)$  of one of the three types discussed earlier, such that  $\arg K(j\omega_k) = \psi_i$  and  $\arg K(j\omega_l) = \psi_j$ . If these phase constraints are infeasible, set  $F(i, j) = \infty$  and consider the next phase pair  $(\Phi_k(i), \Phi_l(j))$ .
  - (c) Find the minimum value of gain  $k_o > 0$  such that (i)  $L_o(j\omega_q) = k_o K(j\omega_q) G(p_o, j\omega_q) \in R_q \cup S_q$  for all  $q = 1, 2, \dots, N$  and (ii)  $L_o(j\omega)$  satisfies Nyquist's encirclement criterion. If no such gain  $k_o$  exists, set  $F(i, j) = \infty$  and consider the next phase pair  $(\Phi_k(i), \Phi_l(j))$ .
  - (d) Calculate the value of the objective function (cross-over frequency, closed-loop bandwidth, asymptotic gain, etc) corresponding to the designed  $L_o(j\omega)$  and assign it to the  $(i, j)$ th element of  $F$ . Save also the controller parameters to the corresponding entries of the parameter arrays.
6. Calculate  $\gamma_o = \min_{(i,j) \in \Phi_k \times \Phi_l} (F)$  and  $(i^*, j^*) \in \operatorname{argmin}(F)$ . If  $\gamma_o = \infty$  the QFT constraints are infeasible; otherwise the optimal cost is  $\gamma_o$  and the

optimal controller parameters can be obtained from the  $(i^*, j^*)$ th elements of the controller-parameter arrays.

A few remarks on the algorithm:

- In step (1) of the algorithm the phase discretisation of the interval  $(-360^\circ, 0^\circ]$  results in a phase grid  $\Phi$ , typically equally spaced. In practice, 50 – 100 phases are adequate. It is helpful to calculate the performance bounds (“Horowitz and “disturbance-rejection” templates) over the same phase grid.
- In principle any two frequencies  $\omega_k$  and  $\omega_l$  can be selected from the set of design frequencies in step (2). Selecting these frequencies reasonably far-apart (for minimum numerical sensitivity) works well in practice.
- In steps (3) and (5a) of the algorithm all phase calculations can be performed modulo  $-360^\circ$ . This restricts the phase interval of interest to the range  $(-360^\circ, 0^\circ]$ .
- Since the phase of  $L_o(j\omega)$  is completely determined when two controller phases are fixed, the calculation of the minimum gain in step (5c) is straightforward. Checking the total number of encirclements required for stability is also straightforward and can be performed by purely graphical means (i.e. by counting the crossings of the  $-180^\circ$  line and their directions). Note also, that a frequency grid “denser” than the set of design frequencies must be used for this purpose.
- Step 5(c) requires the calculations of the performance bounds at arbitrary phases, which may not coincide with the discretised phases of array  $\Phi$ . There is no difficulty, however, in estimating the performance gains from adjacent phase points, e.g. using linear interpolation. Alternatively, the performance bounds may be calculated exactly at these phases to arbitrary accuracy using a bisection algorithm implemented between steps (5b) and (5c).

## 4. Example

Due to lack of space it is not possible to include an example in this paper. A number of numerical examples illustrating the method will be included at the conference presentation.

## 5. Conclusions

An algorithm for the control design of highly-uncertain systems has been outlined. The algorithm is simple, easy to implement, and can be used to automate the loop-shaping step of the QFT design procedure. It can be used to design robust-performance optimal controllers of a simple structure (PID, phase-lead/lag, second-order), or more complex controllers involving arbitrary interconnections of these structures. Extensions of the method to multivariable systems is possible using the standard QFT approach [38, 57].