Control System Design by Convex Optimization

Doctor of Philosophy

in

Electronic Engineering

J. Andrew Donnellan

Dublin City University

School of Electronic Engineering

Supervised by Dr. Anthony M. Holohan

July 2000

Abstract

This thesis deals with the Boyd-Barratt paradigm for feedback controller design. The Boyd-Barratt approach combines the Youla parameterization with convex optimization. In this thesis, their paradigm is accepted in its entirety, but a completely different numerical approach is adopted. An algorithm due to Akilov and Rubinov, which is in essence an abstract rendition of one of the famous algorithms of Remez, is used instead. This completely circumvents the need to compute derivatives or subdifferentials, which can be a difficult task. Instead, certain linear functionals must be computed, and this is generally quite straightforward. An attractive feature of the approach is that the code is much shorter and more elegant. The Boyd-Barratt paradigm has the disadvantage that an infinite dimensional Banach space must be truncated to a finite dimensional subspace prior to optimizing. This thesis also applies certain primal-dual techniques from functional analysis to study the implications of this truncation. Primal-dual theory is used to show that the true optimal solution lies within the solution of two semi-infinite linear programming problems, namely the dual problem with finitely many variables and the primal problem with finitely many variables. Also, it is shown that the alignment property is closely related to the cost of truncation. These results provide an analysis of the effect of truncation.

I hereby certify that this material, which I now submit for assessment on the program for study leading to the award of Doctor of Philosophy in Electronic Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: J. Andrew Donnellan ID No. : 93701276

Dated: 12-7-00

Acknowledgements

I wish to thank my mother and father for their never ending support throughout my life. Their encouragement extends way beyond this thesis. I would also like to thank my brothers and sister for their help.

I must thank my closest friends Jono, John, Sean, Andre, Peter and Bernard. Their never ending encouragement and constructive criticism have played a very important role in the completion of this project.

Finally, I would very sincerely like to thank my supervisor Dr. Anthony Holohan who provided me with excellent guidance and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The design of linear time-invariant (LTI) controllers for LTI plant models which meet given specifications is still a major challenge for control theorists and practitioners.

In the single-input single-output (SISO) case this can usually be done very effectively by using various traditional techniques. These techniques include the root locus method, and methods based around Nyquist, Nichols and Bode plots. The design of controllers for multi-input multi-output (MIMO) systems is quite another matter.

Analytical methods (such as LQG, H_∞ optimal controller design) use an objective functional which when minimized can be used to find a controller. Major disadvantages of analytical techniques are the limited specifications which can be handled, and the difficulty in choosing weights.

The use of parameter optimization techniques, to determine a controller (which is dependent on one or more variable parameters) is one possible approach. A good controller will stabilise the plant and meet certain performance specifications. Optimization methods can handle a much wider range of specifications. However there are serious difficulties with this approach because of the absence of convexity. Boyd and Barratt used the Youla parameterisation to write specifications in a convex format. Through this technique their method could deal with a wide range of specifications.

Unfortunately Boyd and Barratt's technique has two main disadvantages. The first is that it requires a huge software undertaking. The second is that a certain infinite dimensional vector space must be truncated to a finite dimensional subspace, in order to obtain a finite dimensional optimization problem.

This thesis is an effort to improve on their work by tackling both these issues. An alternative numerical approach is proposed. This approach will be shown to require simpler and more standard software. The second disadvantage is analysed using functional analysis techniques. Specifically a qualitative and a quantitative evaluation of the truncation issue is presented.

Chapter 2 describes the Boyd and Barratt paradigm, its range of applicability, and its advantages and disadvantages. Chapter 3 shows how several typical control system specifications can be cast as infinite linear programs. It discusses truncations of the infinite linear programs. Chapter 4 describes in detail an algorithm due to Akilov and Rubinov. Its convergence, mathematical properties and limitations are presented. Chapter 5 describes the author's implementation of the algorithm as well as its performance and validation procedures adopted. Chapter 6 deals with the duality theory of linear programming. Bounds interrelating various infinite, semi-infinite and finite linear programs are established and discussed. Chapter 7 uses techniques from functional analysis to tackle the primary shortcoming of the Boyd and Barratt approach, namely the truncation issue. A qualitative and quantitative assessment of this issue is developed.

Chapters 3, 5 and 7 contain the original contributions of the author, while most of the remainder is based on the literature at large.

Chapter 2

The Boyd and Barratt Paradigm

This chapter deals with the various components of the Boyd and Barratt Paradigm [4,5,6]. The fundamental concept of convexity is outlined. The essential contribution of Boyd and Barratt over previous optimization approaches is described. The breadth of applicability of the approach is described.

2.1 Optimization and Convexity

We begin with two definitions.

A real valued function F is convex if, for all $\alpha \in (0,1)$

$$F(\alpha x_1 + (1-\alpha)x_2) \leq \alpha F(x_1) + (1-\alpha)F(x_2)$$

A set A is convex if, for all $\alpha \in (0,1)$

$$a_1, a_2 \in A \quad \Rightarrow \quad \alpha a_1 + (1-\alpha)a_2 \in A$$

Convexity implies that a local minimum is a global minimum [22,23]. Non-convex problems may have a minimum (or minima) which is a local minimum, but is not a global minimum. There are methods (such as steepest descent) which are guaranteed to converge to a local minimum [22]. Combining the fact that a local minimum is a global minimum for a convex problem with the fact that there are methods to find a local minimum, it follows that a convex problem can be solved with the solution converging to a global minimum. This effectively means that convex problems can be reliably solved numerically, while (most) non-convex problems cannot.

2.2 Direct Controller Optimization

Direct controller optimization techniques are methods to choose the controller, where the controller has some free parameters. For instance direct controller optimization techniques could be used in the case of a PID controller with transfer function given by

$$K(s) = \alpha_p + \alpha_d s + \frac{\alpha_i}{s}$$

or perhaps a controller with a transfer function of

$$K(s) = \frac{\alpha_1 s + \alpha_2}{\alpha_3 s^2 + \alpha_4 s + \alpha_5}$$

This method then uses numerical optimization software to seek the best values of the α's. The specifications are combined into an objective function to be minimized.

Direct controller optimization has the advantage that it can directly handle a wide range of specifications. There are however a few difficulties with these techniques. The solution obtained in this way may be a local minimum as opposed to a global minimum. It can be quite complicated to code such procedures. Also, in order for the procedure to be practical the controller must only have finitely many parameters.

Consider the system shown below in Figure 2.1. Here $G(s)$ is the plant to be controlled, and $K(s)$ is the controller. The following relationships are obtained

$$\begin{pmatrix} e_1 \\ e_2 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} (I+KG)^{-1} & -(I+KG)^{-1}K \\ G(I+KG)^{-1} & (I+GK)^{-1} \\ G(I+KG)^{-1} & -G(I+KG)^{-1} \\ KG(I+KG)^{-1} & K(I+GK)^{-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

It can clearly be seen that all closed loop transfer functions are non-linear in K. Also, all closed looop transfer functions are non-convex in K.

All controller designs must have closed loop stability as a constraint. The sta-

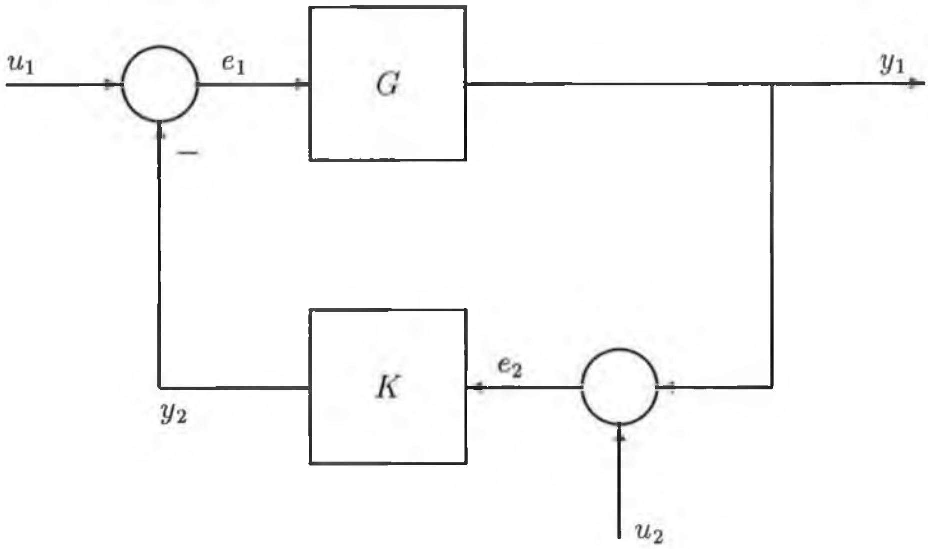

Figure 2.1: A standard feedback system

bility property is non-convex in K and hence cannot always be solved effectively by numerical techniques. As well as the stability property it is also necessary to impose performance specifications. These performance specifications are generally non-convex in the controller K. Thus, this is a serious problem with direct controller optimization.

2.3 Convexity and the Youla parameterisation

Boyd and Barratt's paradigm overcame this problem of direct controller optimization by transforming the problem. The reason the transformed problem is easier to solve is because it is convex. The transformation involved is called the Youla parameterisation, and sometimes the YBJ or YBJK parameterisation, after [28,29]

The Youla Parameterization is an elegant closed form expresssion for all LTI stabilizing controllers. Work done by Youla *et al.* gave the following theorem.

Theorem Let H^∞ denote the set of all stable transfer functions. Let

$$G = D^{-1}N = N_r D_r^{-1}$$

where $N, D, N_r, D_r \in H^\infty$. The above identity is a stable coprime factorisation for the plant. Let

$$\widehat{K} = Y^{-1}X \qquad X, Y \in H^\infty$$

be any one stabilizing controller. Then, all stabilizing controllers are given by

$$K = (Y - QN)^{-1}(X + QD)$$

as Q ranges over H^∞ □

An alternative and equivalent statement is as follows.

Theorem Let H^∞ denote the set of all stable transfer functions. Let

$$G = D^{-1}N = N_r D_r^{-1}$$

where $N, D, N_r, D_r \in H^\infty$. The above identity is a stable coprime factorisation for the plant. Then, all linear time-invariant stabilising controllers are given by

$$K = Y^{-1}X$$

where

$$X = U + QD$$

$$Y = V - QN$$

as Q ranges over all H^∞, and where $U,\ V \in H^\infty$ obey

$$UN + VD = I$$

□

The equation $UN + VD = I$ is known as the Bezout identity.

This theorem means that if Q is viewed as the design variable, then stability is automatically guaranteed. Another crucial observation here is that the behaviour of the closed loop system depends on Q in a much simpler way than it did on K. For example, for the system shown above in Figure 1, the following relationships are obtained.

$$\begin{pmatrix} e_1 \\ e_2 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} D_r(Y-QN) & -D_r(X+QD) \\ N_r(Y-QN) & I-N_r(X+QD) \\ N_r(Y-QN) & -N_r(X+QD) \\ I-D_r(Y-QN) & D_r(X+QD) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

It can clearly be seen that the closed loop transfer functions which were non-linear and non-convex in the controller are affine and convex in the Youla parameter. The main consequence of this is that a wide range of interesting controller optimization problems can be reliably solved numerically. i.e. those of the form:
Minimize a convex objective function subject to (i) closed loop stability, and (ii) convex constraints.

Some examples of the wide range of specifications which can be treated by this approach are given in the next section.

2.4 Convex Specifications

Any specification which can be cast in a form which is convex in $Q(s)$, the Youla parameter, can in theory be solved numerically by the Boyd-Barratt approach. Hence any specification which is convex is a legitimate specification as far as this approach is concerned.

Specifically, the following are convex constraints and hence can be used in the proposed method. Combinations of these specifications can also be treated by this approach.

Sensitivity reduction : The specification

$$||WS||_\infty \leq \lambda$$

is a convex constraint. Here, $S(s)$ is the sensitivity function and reducing it results in reducing the effect of plant uncertainty on the overall closed loop transfer function as well as giving disturbance attenuation.

Complementary sensitivity reduction : The specification

$$\|VT\|_\infty \leq \lambda$$

is also a convex constraint. Reducing the complementary sensitivity has the effect of reducing the effect of measurement noise on the output, as well as improving stability robustnesss.

Asymptotic tracking and regulation : Examples of asymptotic tracking specifications are as follows.

The step response from some command input to some regulated variable must converge to one. This constraint can be expressed as a single linear equality constraint and is a convex constraint. Asymptotic tracking specifications for multivariable systems can be handled as two or more linear equality constraints. Asymptotic regulation and asymptotic decoupling are similar constraints. It may be required, for instance, that a regulated variable asymptotically reject constant inputs.

Closed loop decoupling : It may be required that certain (usually off-diagonal) entries of a closed loop transfer function are to be zero, so that certain inputs have no effect on certain outputs. This constraint is convex.

Overshoot, undershoot and settling time : It may be required to keep the step response between specified limits. A specification of this form can be expressed as a collection of linear inequalities. These constraints are convex and therefore of the desired form.

Bounds on closed loop signal peaks : It may be required that each regulated variable be bounded by some given maximum. This requirement could arise from the requirement not to saturate an actuator or sensor or exceed some internal variable force, torque or current limit. Again the constraint for this is a convex constraint.

Robust disturbance attenuation problem : The robust disturbance attenuation problem is

$$\inf_{Q \in H^\infty} \| |WS| + |VT| \|_\infty$$

This type of problem, while more difficult than the sensitivity reduction problem above can be tackled by convex optimization.

Two disk problem : Another problem is where one wants to limit or minimize

$$\inf_{Q \in H^\infty} \| \max\{|WS|, |VT|\} \|_\infty$$

Again this a convex specification.

Slew rate limitation : A slew rate limitation on the step response can also be cast as a convex constraint.

2.5 Non-Convex Specifications

Some well-known and desirable specifications in control are non-convex. Therefore, it is not possible using this approach to directly optimize such specifications. A general rule of thumb is that specifications on open loop transfer functions cannot be handled using this approach.

Single-loop gain margin and phase margin : The phase margin and gain margin are not convex constraints. Therefore it is not possible using this approach to directly optimize the gain or phase margin. Although it is not possible to do so directly, it is possible to specify them indirectly using M-circle specifications which are convex .

Open loop decoupling : The constraint that K be a 2×2 diagonal matrix is non-convex in Q.

Loop integrity : The constraint that K be a stable transfer function itself, i.e. $K \in H^\infty$, is non-convex in Q.

Controller Complexity : The constraint that K be, for example, a controller with fewer than seven poles is non-convex in Q. Also, for example, PID design cannot be dealt with using this approach.

It has been shown that there are a wide range of specifications which can be effectively treated by the approach of Boyd and Barratt. These specifications can be in both the time domain and the frequency domain.

Chapter 3

Recasting Control Problems as Linear Programs

The Boyd-Barratt paradigm requires that problems are expressed as convex specifications. In this chapter some of the convex specifications listed in the previous chapter are reformulated. They are expressed as infinite linear programs. Truncation of these linear programs is considered.

3.1 Infinite Linear Programs

An infinite linear program is the problem of minimizing a linear cost function

$$c^T x$$

subject to an infinite set of linear inequalities

$$Ax \leq b$$

where the vector x has infinitely many entries.

So in the case of an infinite linear program there are infinitely many variables and infinitely many constraints. So the matrix A is an $\infty \times \infty$ matrix.

A semi-infinite linear program can arise by taking a finite number of variables or a finite number of constraints. If the infinite problem is truncated so that only finitely many variables (FMV) are taken the problem becomes semi-infinite. It has infinitely many constraints but only finitely many variables. In such a case the A matrix is an $\infty \times n$ matrix, where n is the finite number of variables selected.

If the infinite problem is truncated so that only finitely many constraints (FMC) are taken the problem becomes semi-infinite. It has infinitely many variables but only finitely many constraints. In such a case the A matrix is an $m \times \infty$ matrix, where m is the finite number of constraints selected.

If either of the semi-infinite problems were truncated further so that only a finite number of variables and constraints were taken, the resulting problem would be a finite linear program. In this case the A matrix is an $m \times n$ matrix.

3.2 $\|W_1 S\|_\infty$ Specification

Consider the problem of minimizing the weighted sensitivity function,

$$\lambda = \inf_{Q \in H^\infty} \|WS\|_\infty \tag{3.1}$$

This is the original form of the problem and what follows in this subsection is several successive refomulations of this problem. First, the Youla Parameterisation is used to reformulate this problem. The Youla parameterisation makes it possible to convert this problem into a form which is convex in Q, where Q is the Youla parameter. It is necessary to be able to convert the specification into a form which is convex, so that the Boyd-Barratt paradigm can then be applied to it. Before the work done by Youla *et al.* it would not have been possible to treat this specification. This is because the specification is non-convex in the controller K but is convex in Q.

Using the Youla parameterisation as stated in the previous chapter the following expression is obtained for the sensitivity function

$$S = D(QN + V)$$

Substituting the above into (3.1) gives

$$\lambda = \inf_{Q\in H^\infty} \|WD(QN+V)\|_\infty = \inf_{Q\in H^\infty} \|WDQN+WDV\|_\infty$$

Of course, minimizing the weighted sensitivity is not the only H^∞ problem and there are others of interest.

This problem is now of the form: find a vector m belonging to a subspace M which best approximates another given vector x_0,

$$\lambda = \inf_{m\in M} \|m - x_0\|_\infty \tag{3.2}$$

Here, $M = WDNH^\infty$ and $x_0 = -WDV$. This is the second reformulation of the problem. Note that M is an infinite dimensional subspace.

Next, it is shown this problem is an infinite linear program, in infinitely many variables, with infinitely many constraints. Now,

$$\|WS\|_\infty \leq \lambda$$

$$\Longleftrightarrow \|WD(QN+V)\|_\infty \leq \lambda$$

$$\Longleftrightarrow |WDQN(jw) + WDV(jw)| \leq \lambda \ \ \forall\, \omega$$

$$\Longleftrightarrow Re\left\{e^{j\theta}\left(WDQN(jw) + WDV(jw)\right)\right\} \leq \lambda \ \ \forall\, \omega, \theta$$

Let $\{z_i, i \in I\}$ be a basis for H^∞. This set has infinitely many elements. Letting $Q = \sum_{i\in I} \alpha_i z_i$ gives

$$\Longleftrightarrow \sum_{i\in I} \alpha_i Re\left\{e^{j\theta} WDNz_i(j\omega)\right\} \leq \lambda - Re\left\{e^{j\theta} WDV(j\omega)\right\} \ \ \forall\, \omega, \theta \tag{3.3}$$

Clearly, the above constraints are linear in the α_i's and there are infinitely many of them, one for each (ω, θ), where $\omega \in R$, and $\theta \in [0, 2\pi)$. This is the third formulation of the problem,

$$\inf_{\alpha} \lambda \quad \text{subject to}$$

$$\sum_{i\in I}\alpha_i Re\left\{e^{j\theta}WDNx_i(j\omega)\right\} \leq \lambda - Re\left\{e^{j\theta}WDV(j\omega)\right\} \quad \forall\, \omega, \theta \tag{3.4}$$

So (3.4) is an infinite linear program, as expected.

This problem is an infinite linear program. The next reformulation involves truncating H^∞. By truncating H^∞, one obtains a semi-infinite linear program. That is, replace the infinite dimensional space H^∞ by a finite dimensional subspace, as follows.

Take Q to be

$$Q = \sum_{i=1}^{n} \alpha_i y_i(s)$$

where

$$y_i = \frac{s^{i-1}}{(s+\omega_c)^n} \qquad i = 1, ..., n$$

Then m is given by

$$m = WDNQ$$

so substituting for Q in the expression for m gives

$$m = WDN\sum_{i=1}^{n}\alpha_i y_i = \sum_i \alpha_i WDNy_i$$

Now letting $x_i = WDNy_i = WDN\frac{s^{i-1}}{(s+\omega_c)^n}$ gives

$$m = \sum_{i=1}^{n}\alpha_i x_i$$

So taking m in this form and substituting into the original problem gives,

$$\|WS\|_\infty \leq \lambda$$

$$\Longleftrightarrow \left\|\sum_{k=1}^{n}\alpha_k x_k - x_0\right\|_\infty \leq \lambda$$

$$\Longleftrightarrow |WS| \leq \lambda \;\; \forall\, \omega$$

$$\Longleftrightarrow Re\left\{e^{j\theta}WS(j\omega)\right\} \leq \lambda \;\; \forall\, \omega, \theta$$

$$\Longleftrightarrow Re\left\{e^{j\theta}\left(\sum_{k=1}^{n}\alpha_k x_k(j\omega) - x_0(j\omega)\right)\right\} \leq \lambda \;\; \forall\, \omega, \theta$$

$$\iff Re\left\{e^{j\theta}\sum_{k=1}^{n}\alpha_k x_k(j\omega)\right\} \le \lambda + Re\left\{e^{j\theta}x_0(j\omega)\right\} \quad \forall\, \omega, \theta$$

$$\iff \sum_{k=1}^{n}\alpha_k Re\left\{e^{j\theta}x_k(j\omega)\right\} \le \lambda + Re\left\{e^{j\theta}x_0(j\omega)\right\} \quad \forall\, \omega, \theta$$

So the problem now is

$$\inf_{\alpha\in R^n} \lambda \text{ subject to}$$

$$\sum_{k=1}^{n}\alpha_k Re\left\{e^{j\theta}x_k(j\omega)\right\} \le \lambda + Re\left\{e^{j\theta}x_0(j\omega)\right\} \quad \forall\, \omega, \theta \tag{3.5}$$

as ω and θ range through $0 \le \omega \le \infty$ and $0 \le \theta < 2\pi$ respectively.

There are finitely many variables $\lambda, \alpha_1, \alpha_2, \ldots\ldots, \alpha_n$ and the above equation involves infinitely many constraints. So this is a semi-infinite linear program.

Consider next taking only finitely many constraints. Taking only finitely many constraints leads to a finite linear program. This reformulation gives,

$$\inf_{\alpha\in R^n} \lambda \text{ subject to}$$

$$\sum_{j=1}^{n}\alpha_j Re\left\{e^{j\theta_i}x_j(j\omega_i)\right\} \le \lambda + Re\left\{e^{j\theta_i}x_0(j\omega_i)\right\} \quad i = 1, 2, \ldots.., p \tag{3.6}$$

Each constraint, say the i^{th}, corresponds to a certain ω and θ, namely (ω_i, θ_i). Let $b_i = Re\left\{e^{j\theta_i}x_0(j\omega_i)\right\}$ and $a_{ij} = Re\left\{e^{j\theta_i}x_j(j\omega_i)\right\}$, so then eqn.(3.6) becomes

$$\sum_{j=1}^{n}\alpha_j a_{ij} \le \lambda + b_i \quad i = 1, 2, \ldots.., p \tag{3.7}$$

or

$$< \alpha, a_i > \le \ \lambda + b_i$$

Note that this problem is of the traditional linear programming type. It is usually written as

$$\min_x c^T x \text{ subject to } \ Ax \le b$$

Here the A, b and c matrices are given by,

$$A = \begin{pmatrix} Re\left\{e^{j\theta_1}x_1(j\omega_1)\right\} & Re\left\{e^{j\theta_1}x_2(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_1}x_n(j\omega_1)\right\} & -1 \\ \vdots & & & \vdots & \vdots \\ Re\left\{e^{j\theta_p}x_1(j\omega_p)\right\} & Re\left\{e^{j\theta_p}x_2(j\omega_p)\right\} & \dots & Re\left\{e^{j\theta_p}x_n(j\omega_p)\right\} & -1 \end{pmatrix} \quad (3.8)$$

where p is the number of constraints.

$$b = \begin{pmatrix} Re\left\{e^{j\theta_1}x_0(j\omega_1)\right\} \\ Re\left\{e^{j\theta_2}x_0(j\omega_2)\right\} \\ \vdots \\ Re\left\{e^{j\theta_p}x_0(j\omega_p)\right\} \end{pmatrix} \quad (3.9)$$

$$x^T = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n & \lambda \end{pmatrix} \quad (3.10)$$

$$c^T = \begin{pmatrix} 0 & \dots & 0 & 0 & 1 \end{pmatrix} \quad (3.11)$$

Note that A is a $p \times (n+1)$ matrix, b is a $p \times 1$ matrix, x is a $(n+1) \times 1$ matrix and c is a $(n+1) \times 1$ matrix. The matrix A is generally tall. By this it is meant that it has more constraints than variables.

3.3 $\|W_2T\|_\infty$ Specification

The problem of minimizing the weighted complementary sensitivity function is quite similar to the previous case. Hence, it is described here only very briefly. The problem is the following.

$$\lambda = \inf_{Q \in H^\infty} \|W_2T\|_\infty$$

As in the previous case this can be formulated as an infinite linear program. Indeed,

$$\|W_2T\|_\infty \leq \lambda$$

Using the Youla parameterisation gives

$$\iff \|W_2N(-QD+U)\|_\infty \leq \lambda$$

$$\iff -\sum_{i\in I}\alpha_i Re\left\{e^{j\phi}W_2NDz_i(j\omega)\right\} \le \lambda - Re\left\{e^{j\phi}W_2NU(j\omega)\right\} \quad \forall\, \omega,\ \phi$$

The problem is to minimize the least upper bound of

$$Re\left\{e^{j\phi}W_2NU(j\omega)\right\} + \sum_{i\in I}\alpha_i Re\left\{e^{j\phi}W_2NDz_i(j\omega)\right\} \quad \forall\, \omega, \phi \text{ over } \sum_{i\in I}\alpha_i z_i \ \in\ H^\infty$$

This can be reduced to a semi-infinite linear program by replacing the infinite dimensional space H^∞ by a finite dimensional subspace. It can then be reduced to a finite linear program by taking only a finite number of constraints. The constraints for the semi-infinite problem are obtained as follows. Let

$$x_{02} = -W_2NU$$

and as in the previous case take Q to be $Q = \sum_{i=1}^{n}\alpha_i y_i(s)$ where $y_i = \frac{s^{i-1}}{(s+\omega_c)^n}$. So in this case m is given by

$$m = W_2DNQ$$

so substituting for Q gives

$$m = W_2DN\sum_{i=1}^{n}\alpha_i y_i = \sum_{i=1}^{n}\alpha_i W_2DNy_i$$

Now letting $x_i = WDNy_i = WDN\frac{s^{i-1}}{(s+\omega_c)^n}$ gives

$$m = \sum_{i=1}^{n}\alpha_i x_i$$

So taking m in this form and substituting into the original problem gives,

$$\min_{\alpha\in R^n} \lambda \text{ subject to}$$

$$\sum_{k=1}^{n}\alpha_k Re\left\{e^{j\phi}x_{k2}(j\omega)\right\} \le \lambda + Re\left\{e^{j\phi}x_{02}(j\omega)\right\} \quad \forall\omega, \phi$$

Each individual constraint, say the i^{th}, corresponds to a certain ω and ϕ, namely (ω_i, ϕ_i).

In the case of the semi-infinite problem there will be infinitely many of these constraints. Keeping only finitely many constraints gives,

$$\sum_{k=1}^{n} \alpha_k Re\left\{e^{j\phi_i} x_{k2}(j\omega_i)\right\} \leq \lambda + Re\left\{e^{j\phi_i} x_{02}(j\omega_i)\right\} \quad i = 1, 2, \ldots.., p$$

3.4 $\| |W_1 S| + |W_2 T| \|_\infty$ Specification

This problem is called the robust disturbance attenuation problem. Once again this problem is an infinite linear program.

The problem at hand is

$$\lambda = \inf_{Q \in H^\infty} \| |W_1 S| + |W_2 T| \|_\infty$$

Again using the fact that

$$S = D(QN + V) \text{ and } T = N(-QD + U)$$

gives

$$\lambda = \inf_{Q \in H^\infty} \| |W_1 D(QN + V)| + |W_2 N(-QD + U)| \|_\infty$$

$$= \inf_{Q \in H^\infty} \| |W_1 DQN + W_1 DV| + | - W_2 NQD + W_2 NU| \|_\infty$$

This problem is an infinite linear program, in infinitely many variables, with infinitely many constraints. To show this, note that

$$\| |W_1 S| + |VT| \|_\infty \leq \lambda$$

$$\Longleftrightarrow \| |W_1 D(QN + V)| + |W_2 N(-QD + U)| \|_\infty \leq \lambda$$

$$\Longleftrightarrow | \ |W_1 DQN(jw) + W_1 DV(jw)| + | - W_2 NQD(jw) + W_2 NU(jw)| \ | \leq \lambda \ \ \forall\, \omega$$

$$\Longleftrightarrow Re\left\{e^{j\theta} \left(W_1 DQN(jw) + W_1 DV(jw)\right)\right\}$$

$$+ Re\left\{e^{j\phi} \left(-W_2 NQD(jw) + W_2 NU(jw)\right)\right\} \leq \lambda \ \ \forall\, \omega, \theta, \phi$$

Again, let $\{z_i, i \in I\}$ be a basis for H^∞. Letting

$$Q = \sum_{i \in I} \alpha_i z_i$$

gives

$$\Longleftrightarrow \sum_{i \in I} \alpha_i Re\left\{e^{j\theta} W_1 D N z_i(j\omega)\right\} - \sum_{i \in I} \alpha_i Re\left\{e^{j\phi} W_2 N D z_i(j\omega)\right\}$$

$$\leq \lambda - Re\left\{e^{j\phi} W_2 N U(j\omega)\right\} - Re\left\{e^{j\theta} W_1 D V(j\omega)\right\} \quad \forall\, \omega, \theta, \phi$$

and this is an infinite linear program. This gives another but equivalent formulation of the problem,

$$\inf_{\alpha \in R^n} \lambda$$

subject to

$$\sum_{i \in I} \alpha_i Re\left\{e^{j\theta} W_1 D N z_i(j\omega)\right\} - \sum_{i \in I} \alpha_i Re\left\{e^{j\phi} W_2 N D z_i(j\omega)\right\} \tag{3.12}$$

$$\leq \lambda - Re\left\{e^{j\theta} W_1 D V(j\omega)\right\} - Re\left\{e^{j\phi} W_2 N U(j\omega)\right\} \quad \forall\, \omega\,, \theta\,, \phi$$

So this is an infinite linear program, as expected. Truncating H^∞ changes it from an infinite linear program to a semi-infinite linear program as follows. First, note that

$$\|\; |W_1 S| + |W_2 T| \;\|_\infty \leq \lambda$$

$$\|\; |W_1 D(QN + V)| + |W_2 N(-QD + U)| \;\|_\infty \leq \lambda$$

$$\Longleftrightarrow \|\; |m_1 - x_1| + |m_2 - x_2| \;\|_\infty \leq \lambda$$

where

$$m_1 \in W_1 DNQ \qquad x_{01} = -W_1 DV$$

$$m_2 \in -W_2 DNQ \qquad x_{02} = -W_2 NU$$

Next, truncate H^∞. That is, replace the infinite dimensional space H^∞ by a finite dimensional subspace, as follows Take Q to be $Q = \sum_{i=1}^{n} \alpha_i y_i(s)$ where $y_i = \frac{s^{i-1}}{(s+\omega_c)^n}$.

Then

$$m_1 = W_1 DN \sum_i \alpha_i y_i = \sum_i W_1 DN \alpha_i y_i$$

Letting

$$x_{i1} = W_1 DN y_i = W_1 DN \frac{s^{i-1}}{(s+\omega_c)^n}$$

gives

$$m_1 = \sum_{i=1}^{n} \alpha_i x_{i1}$$

and

$$m_2 = -W_2 DN \sum_i \alpha_i y_i = \sum_i -W_2 DN \alpha_i y_i$$

and letting

$$x_{i2} = W_2 DN y_i = W_2 DN \frac{s^{i-1}}{(s+\omega_c)^n}$$

gives

$$m_2 = \sum_{i=1}^{n} \alpha_i x_{i2}$$

Hence,

$$\| \, |W_1 S| + |W_2 T| \, \|_\infty \leq \lambda$$

$$\Longleftrightarrow \left\| |\sum_{k=1}^{n} \alpha_k x_{k1} - x_{01}| + |\sum_{k=1}^{n} \alpha_k x_{k2} - x_{02}| \right\|_\infty \leq \lambda$$

$$\Longleftrightarrow \; |W_1 S(j\omega)| + |W_2 T(j\omega)| \; \leq \lambda \;\; \forall \, \omega$$

$$\Longleftrightarrow Re\left\{e^{j\theta} W_1 S(j\omega)\right\} + Re\left\{e^{j\phi} W_2 T(j\omega)\right\} \leq \lambda \;\; \forall \, \omega, \theta, \phi$$

$$\Longleftrightarrow Re\left\{e^{j\theta}(\sum_{k=1}^{n} \alpha_k x_{k1}(j\omega) - x_{01}(j\omega)\right\} + Re\left\{e^{j\phi}(\sum_{k=1}^{n} \alpha_k x_{k2}(j\omega) - x_{02}(j\omega))\right\} \leq \lambda$$

$$\Longleftrightarrow Re\left\{e^{j\theta} \sum_{k=1}^{n} \alpha_k x_{k1}(j\omega)\right\} + Re\left\{e^{j\phi} \sum_{k=1}^{n} \alpha_k x_{k2}(j\omega)\right\}$$

$$\leq \lambda + Re\left\{e^{j\theta} x_{01}(j\omega)\right\} + Re\left\{e^{j\phi} x_{02}(j\omega)\right\} \;\;\; \forall \, \omega \;, \theta, \phi$$

$$\Longleftrightarrow \sum_{k=1}^{n} \alpha_k Re\left\{e^{j\theta} x_{k1}(j\omega) + e^{j\phi} x_{k2}(j\omega)\right\}$$

$$\leq \lambda + Re\left\{e^{j\theta} x_{01}(j\omega)\right\} + Re\left\{e^{j\phi} x_{02}(j\omega)\right\} \;\;\; \forall \, \omega \;, \theta, \phi$$

So the problem now is

$$\min_{\alpha \in R^n} \lambda$$

subject to

$$\sum_{k=0}^{n} \alpha_k Re\left\{e^{j\theta} x_{k1}(j\omega) + e^{j\phi} x_{k2}(j\omega)\right\}$$

$$\leq \lambda + Re\left\{e^{j\theta} x_{01}(j\omega)\right\} + Re\left\{e^{j\phi} x_{02}(j\omega)\right\} \quad \forall\, \omega\ , \theta, \phi$$

as ω, θ and ϕ range through $0 \leq \omega \leq \infty$, $0 \leq \theta < 2\pi$ and $0 \leq \phi < 2\pi$

Again there are finitely many variables $\lambda, \alpha_1, \alpha_2, \ldots\ldots., \alpha_n$ and the above equation involves infinitely many constraints. So this is a semi-infinite linear program.

Taking only finitely many constraints leads to a finite linear program.

$$\min_{\alpha \in R^n} \lambda \text{ subject to}$$

$$\sum_{k=0}^{n} \alpha_k Re\left\{e^{j\theta_i} x_{k1}(j\omega_i) + e^{j\phi_i} x_{k2}(j\omega_i)\right\}$$

$$\leq \lambda + Re\left\{e^{j\theta_i} x_{01}(j\omega_i)\right\} + Re\left\{e^{j\phi_i} x_{02}(j\omega_i)\right\} \quad i = 1, 2, \ldots.., N$$

Each constraint, say the i^{th}, corresponds to a certain ω, θ and ϕ, namely $(\omega_i, \theta_i, \phi_i)$. So by letting

$$b_i = Re\left\{e^{j\theta_i} x_{01}(j\omega_i)\right\} + Re\left\{e^{j\phi_i} x_{02}(j\omega_i)\right\}$$

and

$$a_{ij} = Re\left\{e^{j\theta_i} x_{k1}(j\omega_i) + e^{j\phi_i} x_{k2}(j\omega_i)\right\}$$

the equation becomes

$$\sum_{j=0}^{n} \alpha_k a_{ij} \leq \lambda + b_i \quad i = 1, 2, \ldots.., p$$

3.5 $\max\{\|W_1 S\|_\infty\ ,\ \|W_2 T\|_\infty\}$ Specification

This two-disc specification is quite similar to the previous case. This problem is the following,

$$\lambda = \inf_{Q \in H^\infty} \max\{\|W_1 S\|_\infty\ ,\ \|W_2 T\|_\infty\}$$

As in the previous cases this can be formulated as an infinite linear program. Indeed,

$$\|W_1 S\|_\infty \leq \lambda$$

$$\Longleftrightarrow \sum_{i\in I} \alpha_i Re\left\{e^{j\theta} W_1 D N z_i(j\omega)\right\} \leq \lambda - Re\left\{e^{j\theta} W_1 D V(j\omega)\right\} \quad \forall\, \omega, \theta \tag{3.13}$$

and

$$\|W_2 T\|_\infty \leq \lambda$$

$$\Longleftrightarrow \sum_{i\in I} \alpha_i Re\left\{e^{j\phi} - W_2 N D z_i(j\omega)\right\} \leq \lambda - Re\left\{e^{j\phi} W_2 N U(j\omega)\right\} \quad \forall\, \omega, \phi$$

The problem is to minimize

$$\max\left\{\sum_{i\in I} \alpha_i Re\left\{e^{j\theta} W_1 D N z_i(j\omega)\right\} + Re\left\{e^{j\theta} W_1 D V(j\omega)\right\}\ ,\right.$$

$$\left. Re\left\{e^{j\phi} W_2 N U(j\omega)\right\} + \sum_{i\in I} \alpha_i Re\left\{e^{j\phi} W_2 N D z_i(j\omega)\right\}\right\} \ \forall\, \omega, \theta \text{ or } \phi \text{ over } \sum_{i\in I} \alpha_i x_i \in H^\infty$$

This can be reduced to a semi-infinite linear program by replacing the infinite dimensional space H^∞ by a finite dimensional subspace.

Let

$$x_{01} = -W_1 D V \qquad x_{02} = -W_2 N U$$

$$x_{i1} = W_1 D N y_i = W_1 D N \frac{s^{i-1}}{(s+\omega_c)^n}$$

$$x_{i2} = W_2 D N y_i = W_2 D N \frac{s^{i-1}}{(s+\omega_c)^n}$$

and the semi-infinite linear program is

$$\min_{\alpha\in R^n} \lambda \text{ subject to}$$

$$\sum_{k=1}^{n} \alpha_k Re\left\{e^{j\theta_i} x_{k1}(j\omega_i)\right\} \leq \lambda + Re\left\{e^{j\theta_i} x_{01}(j\omega_i)\right\} \quad \forall\, \omega, \theta,\ \phi$$

and

$$\sum_{k=1}^{n} \alpha_k Re\left\{e^{j\phi_i} x_{k2}(j\omega_i)\right\} \leq \lambda + Re\left\{e^{j\phi_i} x_{02}(j\omega_i)\right\} \quad \forall\, \omega, \theta,\ \phi$$

Each individual constraint, say the i^{th}, corresponds to a certain ω and either θ or ϕ, namely $(\omega_i, \theta_i \text{ or } \phi_i)$.

3.6 Time Domain Specification

The approach of Boyd and Barratt can deal effectively with time domain specifications, as well as frequency domain specifications. For the purposes of illustration the details for a time domain specification will be given in this section. The problem considered below is to minimize the maximum weighted error between the actual and the desired step response. Once again this problem is an infinite linear program. It is significant that the present method deals successsfully with time domain specifications as well as frequency domain specifications.

Thus consider the problem

$$\lambda = \inf_{Q \in H^\infty} \left\| e^{\tau t}(h - h_{desired}) \right\|_\infty$$

$$= \inf_{Q \in H^\infty} \sup_t \left| e^{\tau t}(h - h_{desired}) \right|$$

where the designer selects the parameter τ and the desired step response. Now the step response $h(t)$ is given by

$$h(t) = L^{-1}\left[\frac{1}{s}T\right] \text{ where } T = N(-QD + U)$$

and

$$h_{desired}(t) = L^{-1}\left[\frac{1}{s}T_{desired}\right] \text{ for given } T_{desired}$$

which leads to the following

$$\lambda = \inf_{Q \in H^\infty} \left\| e^{\tau t}\left(L^{-1}\left[\frac{1}{s}(-NQD + NU \quad - \quad T_{desired})\right]\right) \right\|_\infty$$

$$\lambda = \inf_{Q \in H^\infty} \left\| e^{\tau t}\left(L^{-1}\left[-\frac{1}{s}NQD + \frac{1}{s}NU \quad - \quad \frac{1}{s}T_{desired})\right]\right) \right\|_\infty$$

This problem is of the form of finding a vector m belonging to subspace M which best approximates the vector x_0,

$$\lambda = \inf_{m \in M} \|m - x_0\|_\infty$$

where $M = -DNH^\infty$ is infinite dimensional. This is a norm in the space H^∞.

This problem is an infinite linear program, in infinitely many variables, with infinitely many constraints. Indeed,

$$\left\| e^{\tau t}(h - h_{desired}) \right\| \leq \lambda$$

$$\Longleftrightarrow \left| e^{\tau t}(h - h_{desired}) \right| \leq \lambda \quad \forall\, t \geq 0$$

$$\Longleftrightarrow \max_t \left| e^{\tau t} \left(L^{-1} \left[-\frac{1}{s} NQD + \frac{1}{s} NU) \quad - \quad \frac{1}{s} T_{desired} \right] \right) \right| \leq \lambda$$

Let $\{y_i, i \in I\}$ be a basis for H^∞. This set has infinitely many elements.

$$\Longleftrightarrow \left| e^{\tau t} \left(L^{-1} \left[-\frac{1}{s} NQD + \frac{1}{s} NU \quad - \quad \frac{1}{s} T_{desired}) \right] \right) \right| \leq \lambda \quad \forall\, t \geq 0$$

Letting

$$Q = \sum_{i \in I} \alpha_i y_i$$

gives

$$\Longleftrightarrow -\lambda \leq e^{\tau t} \left(\sum_{i \in I} \alpha_i h_i(t) - h_o(t) - h_{desired}(t) \right) \leq \lambda \quad \forall\, t \geq 0$$

where

$$h_i(t) = L^{-1} \left[-\frac{1}{s} N D y_i \right]$$

and

$$h_0(t) = -L^{-1} \left[\frac{1}{s} NU \right]$$

and this is linear in the α_i's.

This leads to another formulation of the problem,

$$\inf_{\alpha \in R^n} \lambda \text{ subject to}$$

$$\sum_{i\in I}\alpha_i\left\{e^{\tau t}h_i(t)\right\}-\lambda\leq e^{\tau t}\left\{h_o(t)+h_{desired}(t)\right\}\quad\forall\, t$$

and

$$-\sum_{i\in I}\alpha_i\left\{e^{\tau t}h_i(t)\right\}-\lambda\leq e^{\tau t}\left\{-h_o(t)-h_{desired}(t)\right\}\quad\forall\, t \tag{3.14}$$

So this is an infinite linear program, as expected.

Truncating H^∞ changes it from an infinite linear program to a semi-infinite linear program. That is, replace the infinite dimensional space H^∞ by a finite dimensional subspace, as follows Take Q to be $Q=\sum_{i=1}^{n}\alpha_i y_i(s)$ where $y_i=\frac{s^{i-1}}{(s+\omega_c)^n}$. So the problem becomes

$$\inf_{\alpha\in R^n}\lambda\quad\text{subject to}$$

$$\sum_{k=1}^{n}\alpha_k\left\{e^{\tau t}h_k(t)\right\}-\lambda\leq e^{\tau t}\left\{h_o(t)+h_{desired}(t)\right\}\ \text{ and}$$

$$-\sum_{k=1}^{n}\alpha_k\left\{e^{\tau t}h_k(t)\right\}-\lambda\leq -e^{\tau t}\left\{h_o(t)+h_{desired}(t)\right\}\quad\forall\, t\geq 0 \tag{3.15}$$

There are finitely many variables $\lambda,\alpha_1,\alpha_2,.......,\alpha_n$ and the above equation invloves infinitely many constraints. So this is a semi-infinite linear program.

Taking only finitely many constraints leads to a finite linear program.

$$\inf_{\alpha\in R^n}\lambda\ \text{subject to}$$

$$\sum_{k=1}^{n}\alpha_k\left\{e^{\tau t_i}h_k(t_i)\right\}-\lambda\leq e^{\tau t_i}\left\{h_o(t_i)+y_{desired}(t_i)\right\}\ \text{ and}$$

$$-\sum_{k=1}^{n}\alpha_k\left\{e^{\tau t_i}h_k(t_i)\right\}-\lambda\leq -e^{\tau t_i}\left\{h_o(t_i)+y_{desired}(t_i)\right\}$$

Each constraint, say the i^{th}, corresponds to a certain t, say t_i. So by letting

$$b_i=e^{\tau t_i}\left\{h_o(t_i)+h_{desired}(t_i)\right\}\ \text{ and}$$

$$a_{ij}=\left\{e^{\tau t}h_j(t)\right\}\quad i=1,2,.....,p$$

the inequalities become

$$\sum_{j=0}^{n}\alpha_j a_{ij}\leq\lambda+b_i$$

$$-\sum_{j=0}^{n} \alpha_j a_{ij} \leq \lambda - b_i$$

3.7 Discussion

One possible approach to controller design would be to solve the finite linear programs developed above. It would be necessary to have some restrictions on the size of the linear programming problems solved in order to keep them practical. The switch from an infinite linear program to a finite linear program by (i) the use of a subspace of H^∞ and (ii) by having a limit on the number of frequency or time values means that the resulting linear programs have solutions which can be practically evaluated. By reducing the problem to a finite linear program in this way it becomes a problem which can be solved using known practical techniques.

It was shown how the problems could be truncated to finite linear programs. The above development suggests one possible approach to solving these (and similar) problems. One could solve a single large finite linear program. The finite linear programs generated above always have a solution for λ large enough. However there is no guarantee on the accuracy of results obtained in this way. There seems to be no reason to be optimistic that the controller produced by this approach will be close to the true optimal controller. Also, there is the difficulty of deciding how large a linear program to select, and how to choose the subset of constraints to be retained. This suggests seeking a more sophisticated approach. Nonetheless, the details worked out above will be used later.

Chapter 4

The Akilov and Rubinov Algorithm

The Akilov and Rubinov algorithm is described in detail. The proof of convergence is given. The mathematical properties of the algorithm are described. The algorithm has two limitations which are explained. The first limitation is settled in this chapter, while the second limitation will be discussed in detail in later chapters.

4.1 Statement of the Algorithm

We begin by defining some notation.

Let X denote a normed linear space. Let M denote a finite dimensional normed linear subspace, with the norm induced by the norm on X. Let x_o denote an element of X that is not in M. Let X_o denote the finite dimensional normed linear space spanned by M and x_0, with the norm induced by the norm on X.

In the Akilov and Rubinov algorithm the idea is to approximate one function by another function. Thus, suppose that X is a function space, i.e. a vector space of functions equipped with some norm.

The problem considered is that of finding a vector (i.e. a function) in M which is closest to some given function x_o. That is,

$$\mu = \min_{m \in M} \|m - x_o\|$$

The idea is to approximate the given function x_0 as well as possible with a function m from a subspace M. A function m which gives the minimum value μ is called a function of best approximation for x_o and the number μ is called the optimal error of approximation.

Note that a bounded linear functional is defined as a mapping

$$f : X \rightarrow R$$

where

$$< f, x > \text{ is finite } \forall\, x$$

and f is linear.

The algorithm for determining the function of best approximation consists of the successive solution of a number of auxiliary problems as follows.

Step 1 (initialisation) Choose n linear functionals $f_1, ..., f_n$ such that

$$\det |f_i(x_j)| \neq 0 \quad i, j = 1, ..., n \tag{4.1}$$

Then an "interpolating function" $m_n \in M$ exists which is determined by

$$f_i(m_n - x_0) = 0 \;\; \forall\, i = 1, ..., n \tag{4.2}$$

This step gives rise to a set of linear equations which when solved gives a first guess m_n for the function. Let $i = n$.

Step 2 (worst case error) Given m_i, find f_{i+1} so that

$$\mu_i = |f_{i+1}(m_i - x_0)| \;\; = \|m_i - x_o\|\,, \; i = n, n+1, ... \tag{4.3}$$

This amounts to finding a linear functional which achieves the norm of $m_i - x_o$. This means that a new constraint f_{i+1} is determined at the previous step and is added to the old constraints to give a larger set of constraints. If the norm is a max norm, this step involves finding the maximum of $|m_i - x_o|$.

Step 3 (add constraints to linear program and solve) Given $f_1, ..., f_{i+1}$, solve

$$\min_{m \in M} \max_{k \leq i+1} |f_k(m - x_0)|$$

Let m_{i+1} denoting the solution and then

$$\lambda_{i+1} = \min_{m \in M} \max_{k \leq i+1} |f_k(m - x_0)| = \max_{k \leq i+1} |f_k(m_{i+1} - x_0)| \tag{4.4}$$

This step can be cast as a finite linear program. The linear program is then solved to give the new value for the approximating function, m_{i+1}.

Step 4 (check) If $\mu_i - \lambda_{i+1} \leq \epsilon$ stop, otherwise, increment i and go to step 2.

This step checks to see if the approximating function is close enough to the actual function and if so it stops and if not it continues.

The way that this algorithm works is that from each m_j it works out a f_j and from this f_j it obtains the next $m_{j+1} \in M$, and so on. If this $m_{j+1} \in M$ is close enough to the desired x_0 the program stops and if it is not close enough then the next f_{j+1} must be evaluated. The algorithm produces a sequence of finite linear programs. At each iteration, one extra constraint is added to the linear program.

4.2 Proof of Convergence

What follows is the proof that the algorithm converges, giving a function of best approximation, m, corresponding to the minimum value of μ.

Recall that the subproblem of Step 2 was as follows.

$$\mu_i = |f_{i+1}(m_i - x_0)| \ = \|m_i - x_0\|$$

Hence, on each iteration Step 2 produces a new additional linear functional obeying the above equation. Obviously,

$$\mu_i > 0 \tag{4.5}$$

as otherwise

$$\|m_i - x_0\| = 0 \quad \Longrightarrow m_i = x_0 \quad \Longrightarrow x_0 \in M$$

Lemma 4.1 $\lambda_n \leq \lambda_{n+1} \leq \lambda_{n+2} \leq \ldots$

Proof

Clearly,

$$\max_{k \leq i} |f_k(m - x_0)| \leq \max_{k \leq i+1} |f_k(m - x_0)|$$

$$\Longrightarrow \min_{m \in M} \max_{k \leq i} |f_k(m - x_0)| \leq \min_{m \in M} \max_{k \leq i+1} |f_k(m - x_0)| \tag{4.6}$$

Recall that Step 3 was

$$\lambda_{i+1} = \min_{m \in M} \max_{k \leq i+1} |f_k(m - x_0)|$$

giving

$$\lambda_i = \min_{m \in M} \max_{k \leq i} |f_k(m - x_0)| \leq \max_{k \leq i+1} |f_k(m_{i+1} - x_0)| = \lambda_{i+1}$$

So (eqn. 4.6) implies that

$$\lambda_i \leq \lambda_{i+1}$$

Hence the λ_i's form a non-decreasing sequence. □

Lemma 4.2 $\lambda_m \leq \mu \leq \mu_m$

Proof

Now

$$\lambda_{i+1} = \min_{m \in M} \max_{k \leq i+1} |f_k(m - x_0)|$$

Define,

$$\|(m - x_0)\|_{i+1} = \max_{k \leq i+1} |f_k(m - x_0)|$$

Since

$$\max_{k \leq i+1} |f_k(m - x_0)| \leq \max_{all\, f} |f_k(m - x_0)| = \|m - x_0\| \;\; \forall \;\; m \in M$$

it is clear that

$$\lambda_i \leq \mu \tag{4.7}$$

Next, it is shown that $\mu \leq \mu_i$. Since

$$\min_{m \in M} \|m - x_0\| \leq \|\hat{m} - x_0\| \;\; \forall \;\; \hat{m} \in M$$

and

$$\mu_i = \|m_i - x_0\|$$

then

$$\mu \leq \mu_i \tag{4.8}$$

So combining (4.7) and (4.8) gives

$$\lambda_m \leq \mu \leq \mu_m$$

as claimed. □

Lemma 4.3 $\|x\|_{p+1}$ *is a norm, where p denotes the iteration number.*

Proof

By definition

$$\|x\|_{p+1} = \max_{k \leq p+1} |f_k(x)|$$

It must be shown that

$$\text{(i)} \quad \|\alpha x\|_{p+1} = |\alpha| \, \|x\|_{p+1}$$

$$\text{(ii)} \quad \|x + y\|_{p+1} \leq \|x\|_{p+1} + \|y\|_{p+1}$$

$$\text{(iii)} \quad x = 0 \Longleftrightarrow \|x\|_{p+1} = 0$$

By definition

$$\|x\|_{p+1} = \max_{k \leq p+1} |f_k(x)|$$

$$\Rightarrow \|\alpha x\|_{p+1} = \max_{k \leq p+1} |f_k(\alpha x)|$$

$$\Rightarrow \|\alpha x\|_{p+1} = \max_{k \leq p+1} |\alpha f_k(x)|$$

since f_k is a linear functional

$$\Rightarrow \|\alpha x\|_{p+1} = \max_{k \leq p+1} |\alpha| \, |f_k(x)|$$

$$\Rightarrow \|\alpha x\|_{p+1} = |\alpha| \max_{k \leq p+1} |f_k(x)|$$

$$\Rightarrow \|\alpha x\|_{p+1} = |\alpha| \, \|x\|_{p+1}$$

This proves (i).

Using the definition again gives

$$\|x + y\|_{p+1} = \max_{k \le p+1} |f_k(x + y)|$$

which from the linearity property of linear functionals gives

$$\|x + y\|_{p+1} = \max_{k \le p+1} |f_k(x) + f_k(y)|$$

which from the triangle inequality gives,

$$\Rightarrow \|x + y\|_{p+1} \le \max_{k \le p+1} (|f_k(x)| + |f_k(y)|)$$

$$\Rightarrow \|x + y\|_{p+1} \le \max_{k \le p+1} |f_k(x)| + \max_{k \le p+1} |f_k(y)|$$

$$\Rightarrow \|x + y\|_{p+1} \le \|x\|_{p+1} + \|y\|_{p+1}$$

proving (ii). Next it will be shown that $x = 0 \Longrightarrow \|x\|_{p+1} = 0$. By definition with $x = 0$,

$$\|x\|_{p+1} = \|0\|_{p+1} = \max_{k \le p+1} |f_k(0)|$$

since each f_k is linear,

$$\Rightarrow \|0\|_{p+1} = \max_{k \le p+1} |0|$$

$$\Rightarrow \|0\|_{p+1} = 0$$

as required. Next consider the converse of the above, $\|x\|_{p+1} = 0 \Longrightarrow x = 0$. Step 1 states that

$$\det |f_i(x_j)| \ne 0 \quad i, j = 1, ...n$$

and that

$$f_i(m_n - x_0) = 0 \;\; \forall \; i = 1, ..., n$$

These facts imply that there exists $\alpha_1, \ldots, \alpha_n$ such that

$$\begin{pmatrix} f_1(x_1) & \ldots & f_1(x_n) \\ \vdots & & \vdots \\ f_n(x_1) & \ldots & f_n(x_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} f_1(x_0) \\ \vdots \\ f_n(x_0) \end{pmatrix}$$

$$\Longrightarrow \sum_{j=1}^{n} \alpha_j f_i(x_j) = f_i(x_0) \quad i = 1, ..., n$$

and since each f_i is linear,

$$\Longrightarrow f_i\left(\sum_{j=1}^{n} \alpha_j x_j\right) = f_i(x_0) \quad i = 1, ..., n$$

$$\Longrightarrow f_i\left(\sum_{j=1}^{n} \alpha_j x_j - x_0\right) = 0 \quad i = 1, ..., n \tag{4.9}$$

Suppose now that $\|x\|_{p+1} = 0$. Then by definition

$$\Longrightarrow \max_{k \leq p+1} |f_k(x)| = 0$$

$$\Longrightarrow |f_k(x)| = 0 \ \ \forall \ \ k = 1, ..., n \tag{4.10}$$

$$\Longrightarrow f_k(x) = 0 \ \ \forall \ \ k = 1, ..., n \tag{4.11}$$

Since $x \in X_0$, and since X_0 is a finite dimensional normed linear space spanned by $x_o, x_1, x_2, ..., x_n$, we may write

$$x = -bx_0 + \sum_{i=1}^{n} c_i x_i \quad \text{for some} \quad b, c_1, ..., c_n \in R$$

Rewriting and letting $c_i = b\alpha_i - a_i$ gives,

$$x = -bx_0 + \sum_{i=1}^{n} (b\alpha_i - a_i) x_i \tag{4.12}$$

and from equation (4.11)

$$\Longrightarrow f_k\left(-bx_0 + \sum_{i=1}^{n} (b\alpha_i - a_i) x_i\right) = 0 \ \ \forall \ \ k = 1, ..., n \tag{4.13}$$

Now using (4.9), gives

$$bf_i\left(\sum_{j=1}^{n}\alpha_j x_j - x_0\right) = 0 \quad i = 1, ..., n$$

$$\Longrightarrow \sum_{i=1}^{n} a_i f_k(x_i) = 0 \;\; \forall \;\; k = 1, ..., n$$

$$\Longrightarrow \begin{pmatrix} f_1(x_1) & \dots & f_1(x_n) \\ \vdots & & \vdots \\ f_n(x_1) & \dots & f_n(x_n) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

but this contradicts the fact that $\det|f_i(x_j)| \neq 0$ unless $(a_1, ...a_n) = 0$. Hence,

$$\Longrightarrow c_i = b\alpha_i$$

and so substituting into (4.12) gives

$$x = -b\left(x_0 - \sum_{i=1}^{n}\alpha_i x_i\right) \qquad \text{for some} \quad b, c_1, ..., c_n \in R$$

which when substituted into eqn. (4.13) implies

$$f_k(x) = 0 \;\; \forall \, k$$

Since x is here any element of $\text{span}\{x_0, x_1, ..., x_n\}$ then $f_k = 0$ contradicting eqn. (4.5) unless $x = 0$. So $\|x\|_{p+1} = 0 \Longrightarrow x = 0$, as required □

It has been shown that at each iteration $\lambda_m \leq \mu \leq \mu_m$ and it will now be shown that these bounds converge to the optimal solution.

Lemma 4.4 *The algorithm produces a subsequence $m_{i_j} \in M$ for which $\mu_{i_j} \to \mu$ and $\lambda_{i_j} \to \lambda$.*

Proof

From the properties of norms on finite dimensional spaces, it is known that

$$\exists \; k > 0 \;\; \text{such that} \;\; \|x\| \leq k\|x\|_{p+1} \;\; \forall \;\; x \in X_0$$

Now,

$$\mu_i = \|m_i - x_0\|$$

so

$$\Rightarrow \mu_i \leq k \|m_i - x_0\|_{p+1}$$

$$\Rightarrow \mu_i \leq k \max_{k \leq p+1} |f_k(m_i - x_0)| \quad i \geq p$$

But

$$\lambda_i = \min_{m \in M} \max_{k \leq i} |f_k(m - x_0)|$$

$$\Rightarrow \mu_i \leq k\lambda_i$$

Using Lemma 4.2

$$\Rightarrow \mu_i \leq k\mu \tag{4.14}$$

and hence μ_i is bounded. Since X_0 is finite dimensional the sequence μ_i must have a convergent subsequence μ_{i_j}.

It can also be shown that m_i is bounded. Indeed, using the triangle inequality

$$\|m_i\| \leq \|m_i - x_0\| + \|x_0\|$$

But,

$$\mu_i = \|m_i - x_0\|$$

and so

$$\|m_i\| \leq \mu_i + \|x_0\|$$

which from equation (4.14) gives,

$$\|m_i\| \leq k\mu + \|x_0\|$$

Again the sequence m_i must therefore have a convergent subsequence m_{i_j}.

Using these subsequences,

$$\mu_{i_j} = \left|f_{i_j+1}\left(m_{i_j} - x_0\right)\right| = \left\|m_{i_j} - x_0\right\|$$

$$\Rightarrow \mu_{i_j} \leq \max_{k \leq i_{j+1}} \left| f_k \left(m_{i_j} - x_0 \right) \right|$$

$$\Rightarrow \mu_{i_j} \leq \max_{k \leq i_{j+1}} \left| f_k \left(m_{i_{j+1}} - x_0 \right) \right| + \max_{k \leq i_{j+1}} \left| f_k \left(m_{i_{j+1}} - m_{i_j} \right) \right|$$

$$\Rightarrow \mu_{i_j} \leq \lambda_{i_{j+1}} + \left\| m_{i_{j+1}} - m_{i_j} \right\|$$

since m_{i_j} is a Cauchy sequence,

$$\mu_{i_j} \leq \lambda_{i_{j+1}} + \epsilon \tag{4.15}$$

From equations (4.7), (4.8), and (4.15),

$$\lambda_{i_j} \leq \mu \leq \mu_{i_j} \leq \lambda_{i_{j+1}} + \epsilon$$

and since m_{i_j} is a Cauchy sequence, we have

$$\epsilon \rightarrow 0 \quad \text{as} \quad j \rightarrow \infty$$

giving

$$\lambda_{i_j} \rightarrow \mu \qquad \lambda_{i_j} \leq \mu \tag{4.16}$$

$$\mu_{i_j} \rightarrow \mu \qquad \mu \leq \mu_{i_j} \tag{4.17}$$

which completes the proof □

4.3 Properties of the Akilov and Rubinov Algorithm

This section describes certain properties of the algorithm.

Convergence

- The algorithm is guaranteed to converge. It converges to a global optimum.
- The λ_i's converge to the optimal value μ from below. The λ_i's form a non-decreasing monotonic sequence.

- The μ_i's converge to μ from above. However the convergence is not necessarily in a monotonic fashion. A subsequence of the μ_i's converge to μ.

- The λ_i's and μ_i's converge towards some final answer μ. At no stage do λ_i and μ_i cross over each other. So at all stages $\lambda_i \leq \mu_i$, and equality holds if and only if m_i is exactly the optimal solution. This important property of the Akilov and Rubinov algorithm means that at each stage an upper and lower bound on μ is obtained. In practice this is very useful because it means that at each iteration one can see how close one is to the optimal answer, μ.

Increasing n

- If the proposed algorithm is executed an upper and lower bound will be obtained for each iteration and hence at termination. The optimal μ lies within these bounds. Suppose now that the number of terms in the subspace of H_∞ spanned by x_i is increased by increasing n then the new range can not be further from the true optimal solution. The interval in which μ lies may have a larger upper limit for a larger n, but this is not in contradiction with the algorithm provided there is still a range in which μ may lie and not do worse for a larger n.

- Having said the above, it is expected that for most cases that increasing the subspace of H_∞ spanned by x_i it should give a range containing μ which is in fact closer to the optimal solution. It can never do worse.

4.4 Applicability to Boyd and Barratt

The Akilov and Rubinov algorithm as stated previously does provide guarantees. Firstly, it is guaranteed to converge. However, there are two requirements needed for application of the algorithm. It is required that (i) the subspace X_0 be finite dimensional and that (ii) the problem can be expressed as a norm. In what follows the norm restriction will be treated. As stated, it is necessary for the algorithm that specifications can be written as a norm. Consider the robust disturbance attenuation problem,

$$\inf_Q \| |W_1 S| + |W_2 T| \|_\infty$$

Define

$$\left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} = \| |u_1| + |u_2| \|_\infty = \sup_{j\omega} |u_1| + |u_2|$$

It is claimed that, this is a norm in the space $\Upsilon = H^\infty \times H^\infty$. It is a straightforward matter to show that the space $\Upsilon = H^\infty \times H^\infty$ is a vector space. Indeed, the direct sum of two vector spaces is a vector space. It is necessary to show that

$$\left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty}$$

is indeed a norm. In order to prove this the following must be shown

$$\text{(i)} \quad \left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} \geq 0$$

$$\text{(ii)} \quad \left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} = 0 \quad \text{if and only if} \quad u = 0$$

$$\text{(iii)} \quad \left\| c \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} = |c| \left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} \quad \text{where c is some scalar}$$

$$\text{(iv)} \quad \|u + v\|_{\infty \times \infty} \leq \|u\|_{\infty \times \infty} + \|v\|_{\infty \times \infty}$$

where $u = (u_1, u_2)$ and $v = (v_1, v_2)$ are elements of the space $\Upsilon = H^\infty \times H^\infty$.

Now,

$$\left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty \times \infty} = \| |u_1| + |u_2| \|_\infty = \sup_{jR} |u_1| + |u_2|$$

but both

$$|u_1| \geq 0 \text{ and } |u_2| \geq 0 \; \forall \, j\omega \in \; jR$$

and so the sum

$$|u_1| + |u_2| \geq 0$$

Hence,

$$\sup_{jR} |u_1| + |u_2| \geq 0$$

Since the supremum over a set of non-negative real values is necessarily non-negative,

$$\|u\|_{\infty\times\infty} \geq 0$$

as required.

Next, consider property (ii) above. Letting $u_1 = 0$ and $u_2 = 0$ in $\|u\|_{\infty\times\infty} = 0$ the following is obtained

$$\left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty\times\infty} = \| |0| + |0| \|_\infty = \sup_{jR} |0| + |0| = \sup_{jR} 0 = 0$$

Conversely, letting $\|u\|_{\infty\times\infty} = 0$ gives the following,

$$\left\| \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \right\|_{\infty\times\infty} = 0$$

$$\Longrightarrow \| |u_1| + |u_2| \|_\infty = 0$$

$$\Longrightarrow \sup_{jR} |u_1| + |u_2| = 0$$

$$\Longrightarrow u_1 = 0 \text{ and } u_1 = 0$$

Next consider (iii). Letting $u = (u_1, u_2)$ gives

$$\|cu\|_{\infty\times\infty} = \| |cu_1| + |cu_2| \|_\infty$$

$$= \| |c||u_1| + |c||u_2| \|_\infty$$

$$= \| |c|(|u_1| + |u_2|) \|_\infty$$

$$= |c| \, \| |u_1| + |u_2| \|_\infty$$

$$= |c| \|u\|_{\infty\times\infty}$$

Finally, consider (iv), which is

$$\|u + v\|_{\infty\times\infty} \leq \|u\|_{\infty\times\infty} + \|v\|_{\infty\times\infty}$$

Now,

$$\|u+v\|_{\infty\times\infty} = \||u_1+v_1|+|u_2+v_2|\|_{\infty} = \sup_{jR}|u_1+v_1|+|u_2+v_2|$$

Using the triangle inequality for complex numbers

$$\leq \sup_{jR}|u_1|+|v_1|+|u_2|+|v_2|$$

$$\leq \sup_{jR}|u_1|+|u_2|+\sup_{jR}|v_1|+|v_2|$$

$$= \|u\|_{\infty\times\infty}+\|v\|_{\infty\times\infty}$$

since the sum of the suprema of two parts separately is greater than the supremum of the sum of the two parts at once. Thus, the desired expression is obtained,

$$\|u+v\|_{\infty\times\infty} \leq \|u\|_{\infty\times\infty}+\|v\|_{\infty\times\infty}$$

Thus it has been shown that $\|u\|_{\infty\times\infty}$ is indeed a norm. It has therefore been shown that the robust disturbance attenuation problem can be expressed as a norm. It could similarly be shown for the two-disc problem.

4.5 Limitations of the Algorithm

The algorithm suffers from two important restrictions

Firstly, it is necessary that the specifications can be written in the form of a norm. It was shown that the specifications could be written in the form of linear constraints. This is enough to attempt to solve the problem as a large finite linear program, but as was stated previously there are no guarantees about the solution obtained in this way. In order to use the Akilov and Rubinov approach, which does provide some guarantees it is necessary to format the specifications as norms. Thus it is necessary that it be shown that $\|u\|_{\infty\times\infty}$, etc are in fact norms.

Secondly, in order to apply the Akilov and Rubinov algorithm it is necessary that H^{∞} be truncated. The algorithm gives no indication of how far the truncated solution is from the true optimum. For example, take the $\|WS\|_{\infty}$ specification and let M denote the finite dimensional subspace of H^∞ that is used in the semi-infinite linear program. Letting

$$\lambda_\infty = \inf_{Q \in H_\infty} \|WS\|_\infty$$

and

$$\lambda_M = \inf_{Q \in M} \|WS\|_\infty$$

then since M is a subspace of H^∞

$$\lambda_\infty \leq \lambda_M$$

An important issue is the gap between λ_∞ and λ_M. How much does the truncation cost? Similar remarks apply to the other specifications.

This is an issue which will be returned to in later chapters.

4.6 Comments

In this chapter a detailed description was given of the Akilov and Rubinov algorithm. A proof of its convergence was given. The properties of the algorithm were then described. Finally some restrictions and limitations of the algorithm were given. It was established that the norm property applies to the robust disturbance attenuation problem, as required by the algorithm. The effects of the latter limitation will be analysed below.

Chapter 5

Software Development and Experience

In this chapter, the choice of the Akilov and Rubinov algorithm within the context of optimal robust controller design is explained. The algorithm as described in the previous chapters has been coded in Matlab 5.2 for the following problems.

$$\|WS\|_\infty \leq \lambda$$

$$\|W_2T\|_\infty \leq \lambda$$

$$\||W_1S| + |W_2T|\|_\infty \leq \lambda$$

$$\max\{\|W_1S\|_\infty, \|W_2T\|_\infty\} \leq \lambda$$

$$\lambda = \inf_{Q \epsilon H^\infty} \left\| e^{\tau t}(h - h_{desired}) \right\|_\infty$$

This chapter discusses the author's implementation of these specifications. How the software was validated is described.

5.1 Why use the Akilov and Rubinov Algorithm?

The optimization approach of Boyd and Barratt can handle a wide range of specifications. They used the Youla parameterisation to write specifications in a convex format, and then used parameter optimization techniques. This thesis proposes using instead an algorithm by Akilov and Rubinov which solves convex problems. The reasons for this choice is outlined here by giving some of the advantages of the Akilov and Rubinov algorithm.

Compatibility The proposed method retains the advantages of the Boyd-Barratt approach. Thus, a wide range of specifications can be treated, and convergence to a global optimum is guaranteed.

Bounds The algorithm has the attractive property that it gives a lower and an upper bound at each iteration.

Convergence Monitoring Using the upper and lower bounds mentioned above it is possible to see how quickly the algorithm is converging. It also makes it easier to decide when to stop, i.e. to decide when all specifications are met to a sufficiently high level. This is less than obvious with descent methods.

Speed It is faster because the code is shorter.

Standard Sub-problems The algorithm requires a linear equation solver and a linear program solver, which are standard numerical problems.

Computional Ease It has the advantage over differential descent methods that it eliminates the need to compute complicated gradients (derivatives, descent directions, etc.). Instead, it requires certain linear functionals which are much easier to determine.

5.2 The Algorithm's Performance

This section contains data obtained from the author's coding of this approach.

As an example, consider the following simple model of a servo,

$$G = \frac{1}{s(s+0.01)}$$

Consider the following specification,

$$\| |W_1 S| + |W_2 T| \|_\infty \leq \lambda$$

where

$$W_1 = \frac{\frac{1}{901}(s+30)^2}{(s+1)^2}$$

$$W_2 = \frac{\frac{1}{901}(s+1)^2}{(s+30)^2}$$

The results obtained are given in Table 5.1. Graphs of this data are given in Figures 5.1 to 5.5.

Case of n = 3				
ϵ	μ	λ	iterations	flops
0.01	0.20117	0.19486	10	288443
0.001	0.19795	0.19744	15	1184685
0.0001	0.19784	0.19775	20	3476658
1e-005	0.1978	0.19779	25	8485807

For case of n = 4				
ϵ	μ	λ	iterations	flops
0.01	0.10275	0.095658	12	8999276
0.001	0.098425	0.097436	17	10454698
0.0001	0.097737	0.097685	22	13937398
1e-005	0.097709	0.097701	24	18696825

Case of n = 5				
ϵ	μ	λ	iterations	flops
0.01	0.070089	0.06208	14	19504019
0.001	0.066318	0.06556	20	21960987
0.0001	0.065674	0.065617	24	26596113
1e-005	0.065653	0.065643	30	37043487

Case of n = 6				
ϵ	μ	λ	iterations	flops
0.01	0.069226	0.061167	16	38340366
0.001	0.06421	0.063216	21	41570336
0.0001	0.063721	0.063643	27	49425768
1e-005	0.063654	0.063648	31	62665109

Case of n = 7				
ϵ	μ	λ	iterations	flops
0.01	0.067056	0.060693	19	65017302
0.001	0.063971	0.062999	24	70334913
0.0001	0.063472	0.063422	32	85386242
1e-005	0.063439	0.063432	37	111594020

Case of n = 8				
ϵ	μ	λ	iterations	flops
0.01	0.069996	0.060533	20	114502413
0.001	0.063961	0.063032	28	124217802
0.0001	0.063395	0.06331	35	146004155
1e-005	0.063338	0.063329	42	189812870

Table 5.1: Results for the robust disturbance attenuation problem applied to a servo motor.

Using the same model and weighting functions, results were also obtained for the two-disc problem,

$$\inf_{Q \epsilon H^{\infty}} \max \{ \|W_1 S\|_{\infty}, \|W_2 T\|_{\infty} \}$$

The results obtained are given in Table 5.2. Graphs of this data are given in Figures 5.6 to 5.9.

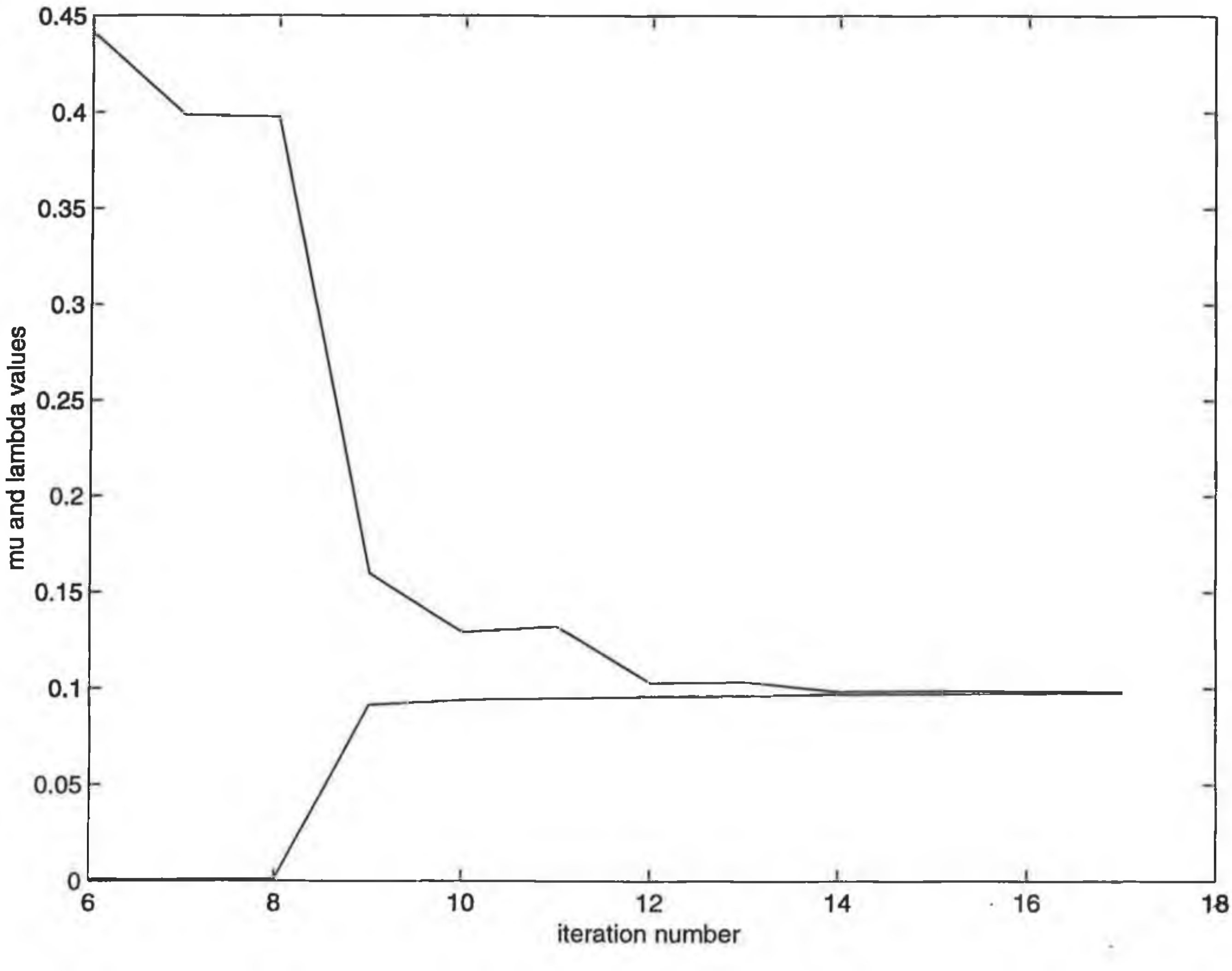

Figure 5.1: μ and λ values vs. iteration number for $\epsilon = 0.001$ and $n = 4$ for RDAP example

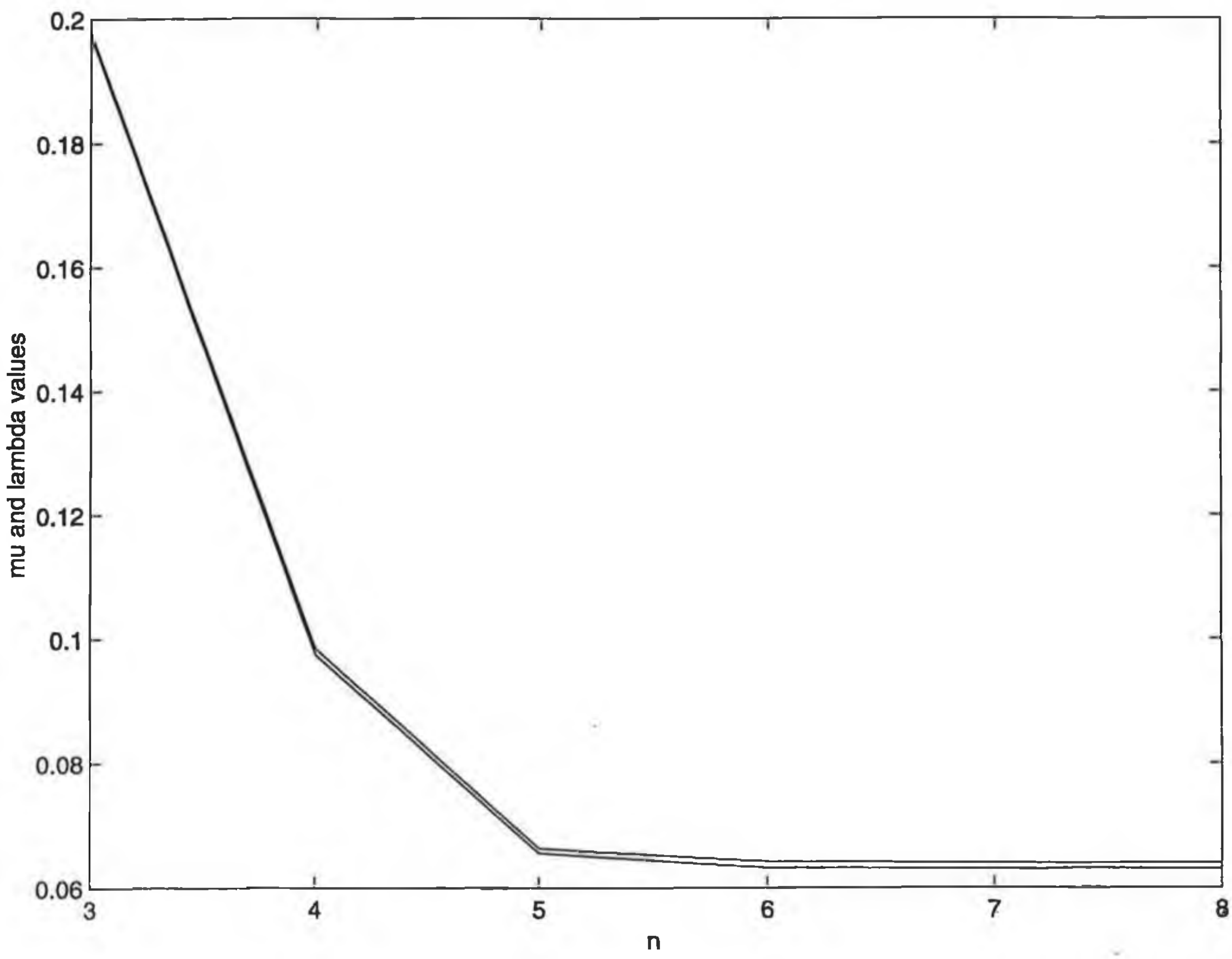

Figure 5.2: μ and λ values vs. n, $\epsilon = 0.001$ for RDAP example

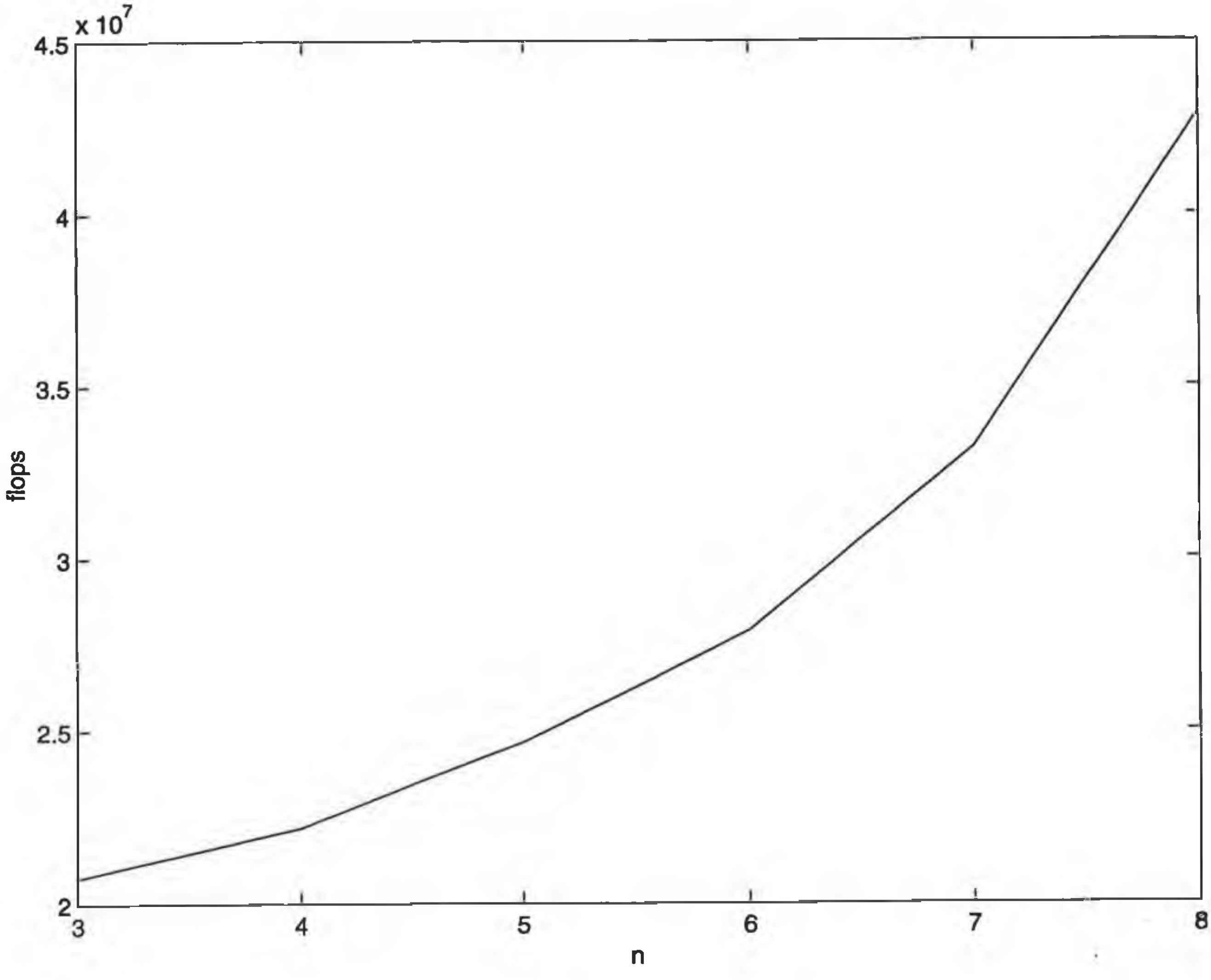

Figure 5.3: flops vs. n, $\epsilon = 0.001$ for RDAP example

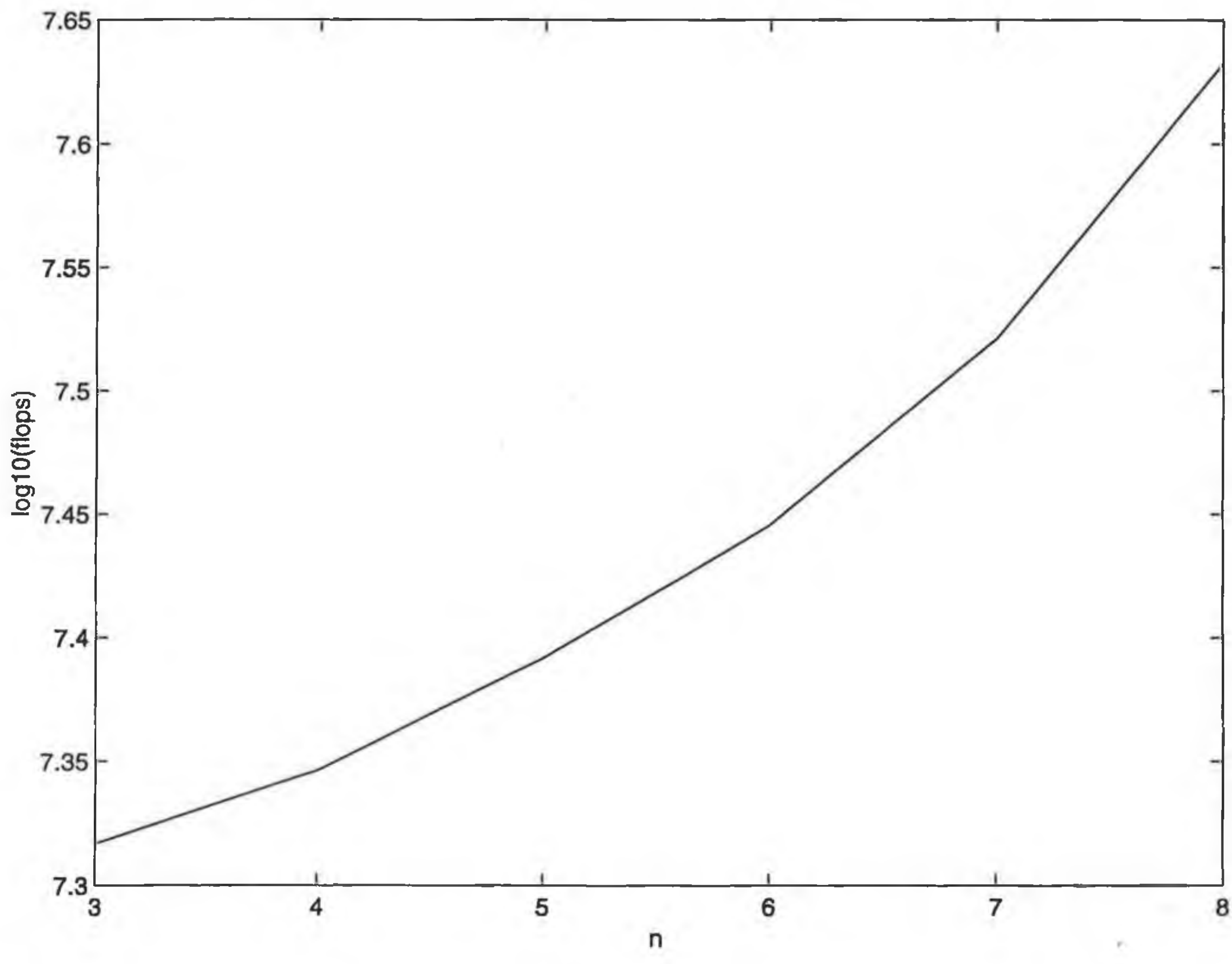

Figure 5.4: $log_{10}(flops)$ vs. n, $\epsilon = 0.001$ for RDAP example

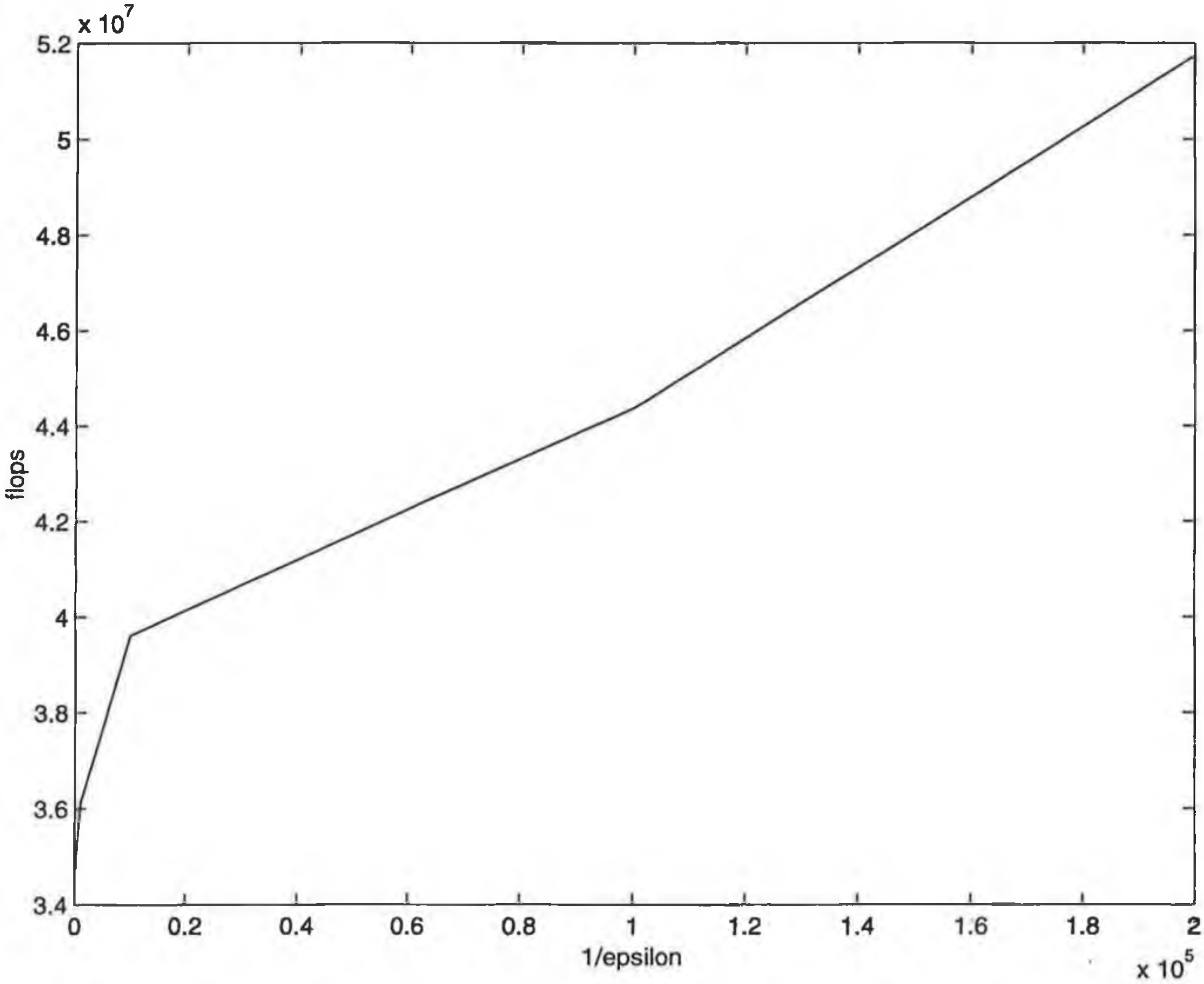

Figure 5.5: flops vs. $\frac{1}{\epsilon}$, $n = 4$ for RDAP example

Case of n = 3				
ϵ	μ	λ	iterations	flops
0.01	0.18985	0.18645	13	199028361
0.001	0.18809	0.18737	15	199812504
0.0001	0.18768	0.18763	20	201997225
1e-005	0.18765	0.18764	23	205639596

Case of n = 4				
ϵ	μ	λ	iterations	flops
0.01	0.091027	0.084775	12	206036193
0.001	0.089441	0.088596	15	206852388
0.0001	0.088972	0.088879	19	208693539
1e-005	0.088932	0.088925	24	212935877

Case of n = 5				
ϵ	μ	λ	iterations	flops
0.01	0.061263	0.056445	14	213640368
0.001	0.060165	0.059226	18	215278072
0.0001	0.059531	0.059475	23	219151704
1e-005	0.059496	0.059491	28	227654033

Case of n = 6				
ϵ	μ	λ	iterations	flops
0.01	0.064125	0.05442	17	229134925
0.001	0.058069	0.057255	23	233458051
0.0001	0.05774	0.057671	28	242337686
1e-005	0.057713	0.057708	33	259183073

Case of n = 7				
ϵ	μ	λ	iterations	flops
0.01	0.061914	0.05544	19	261427005
0.001	0.057855	0.057128	24	266573234
0.0001	0.057523	0.057438	30	278277863
1e-005	0.057472	0.057469	37	304953375

Case of n = 8				
ϵ	μ	λ	iterations	flops
0.01	0.06444	0.054938	19	307313600
0.001	0.05777	0.056787	27	315560404
0.0001	0.057445	0.057364	33	332771686
1e-005	0.05738	0.057375	41	372040579

Table 5.2: Results for the two-disc problem applied to a servo motor.

The paradigm was also tested on the time domain specification

$$\lambda = \inf_{Q \epsilon H^{\infty}} \left\| e^{\tau t}(h - h_{desired}) \right\|_{\infty}$$

The results obtained are given in Table 5.3. Graphs of this data are given in Figures 5.10 to 5.12.

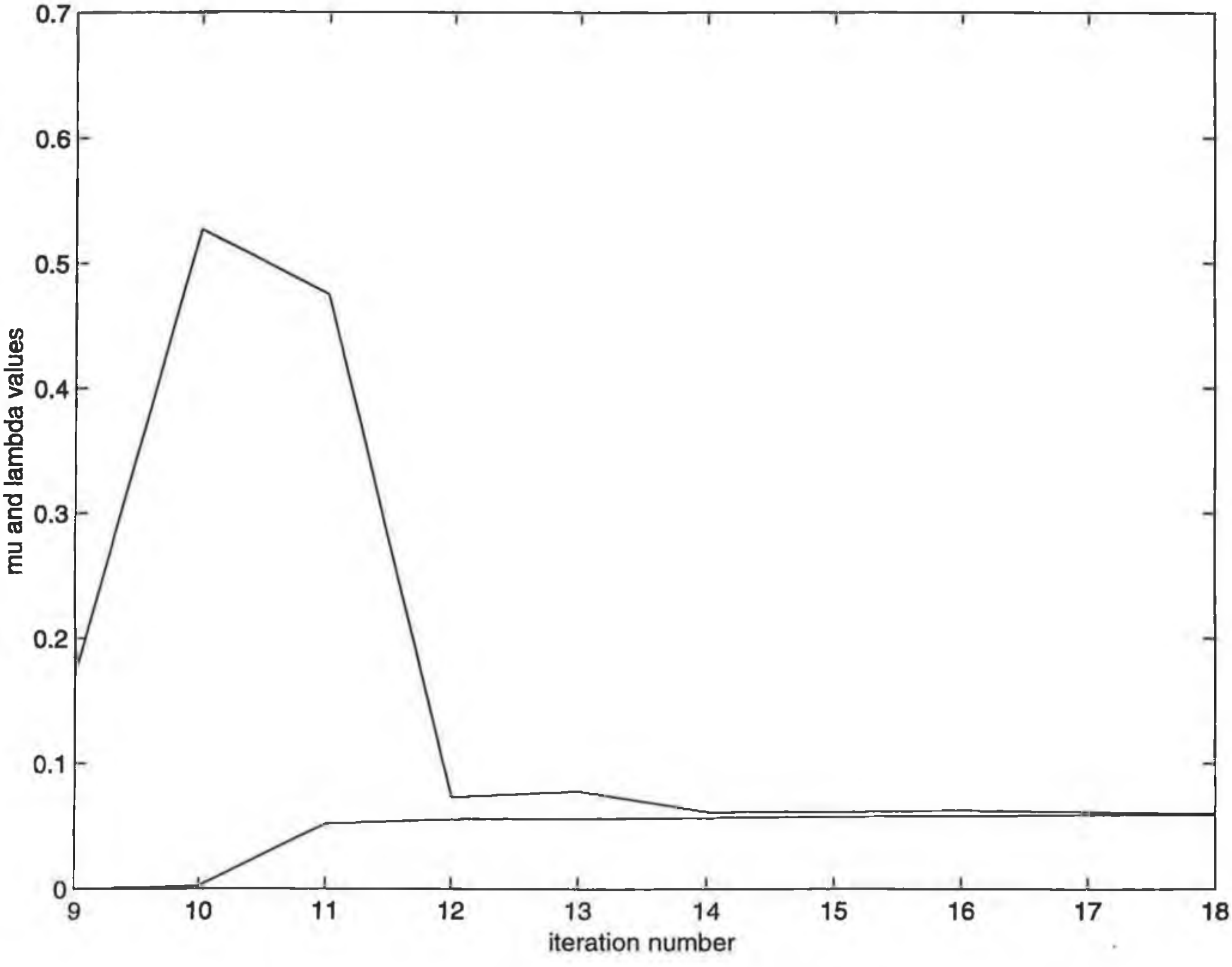

Figure 5.6: μ and λ values vs. iteration number for $\epsilon = 0.001$ and $n = 4$ for Two Disc example

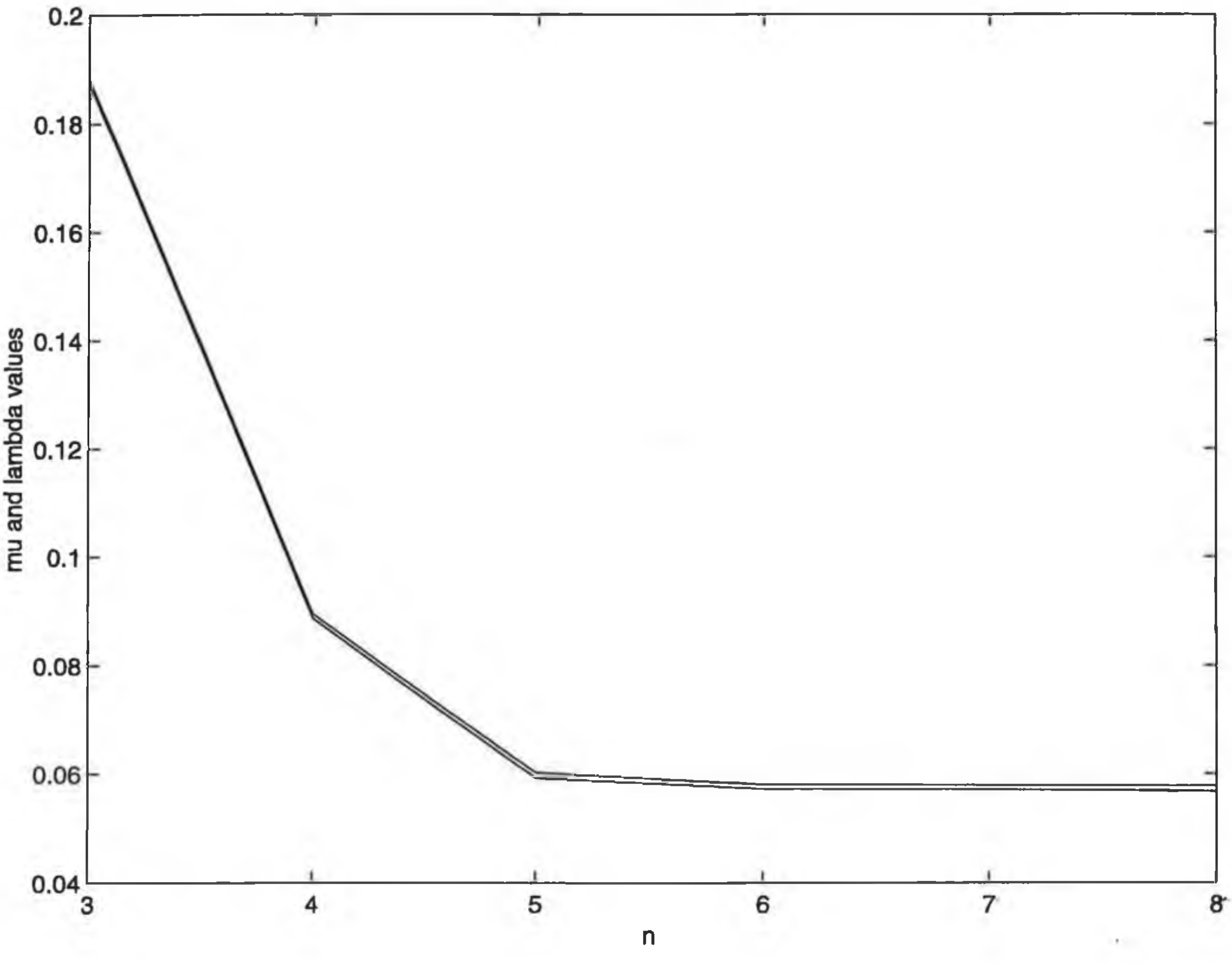

Figure 5.7: μ and λ values vs. n, $\epsilon = 0.001$ for Two Disc example

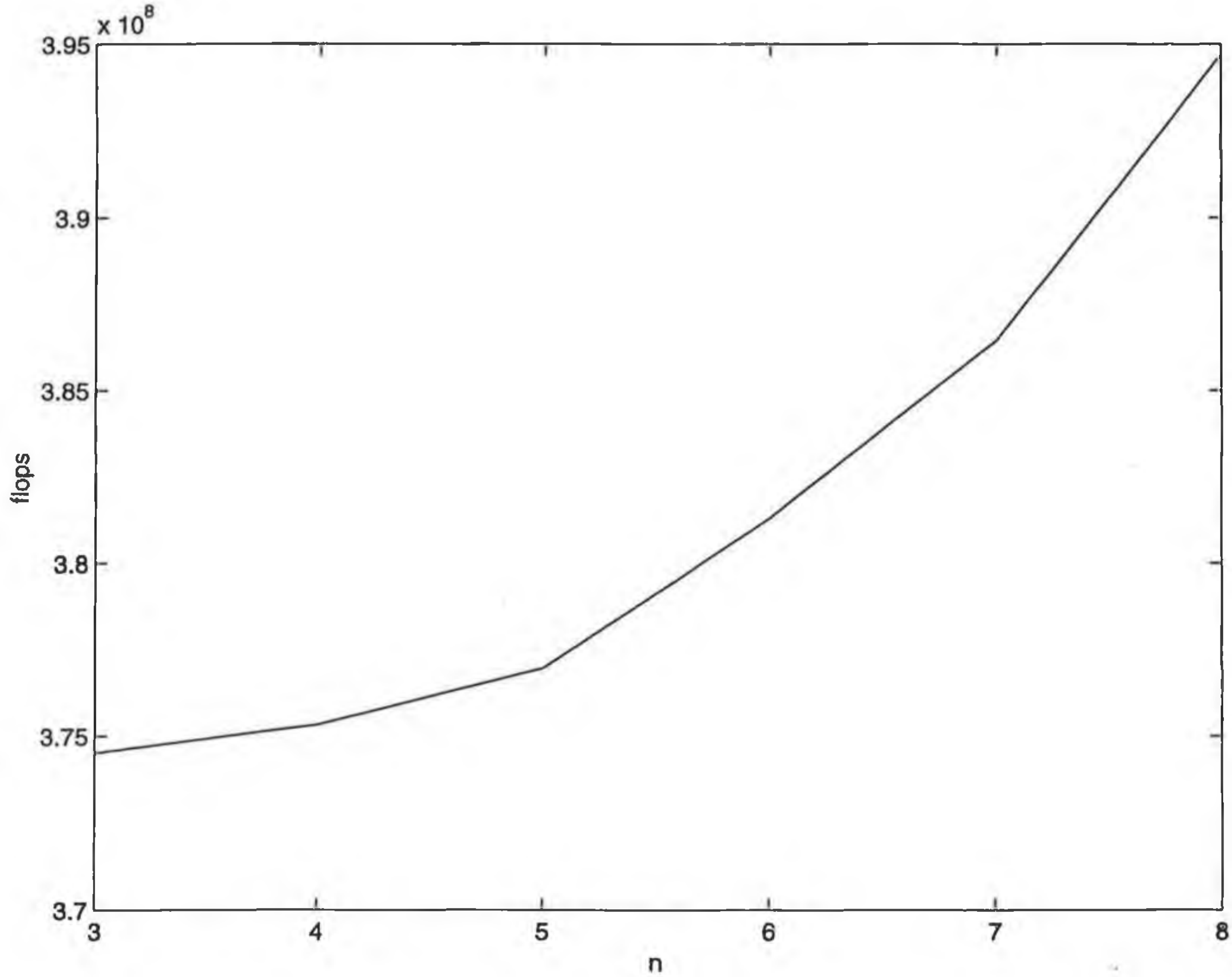

Figure 5.8: flops vs. n, $\epsilon = 0.001$ for Two Disc example

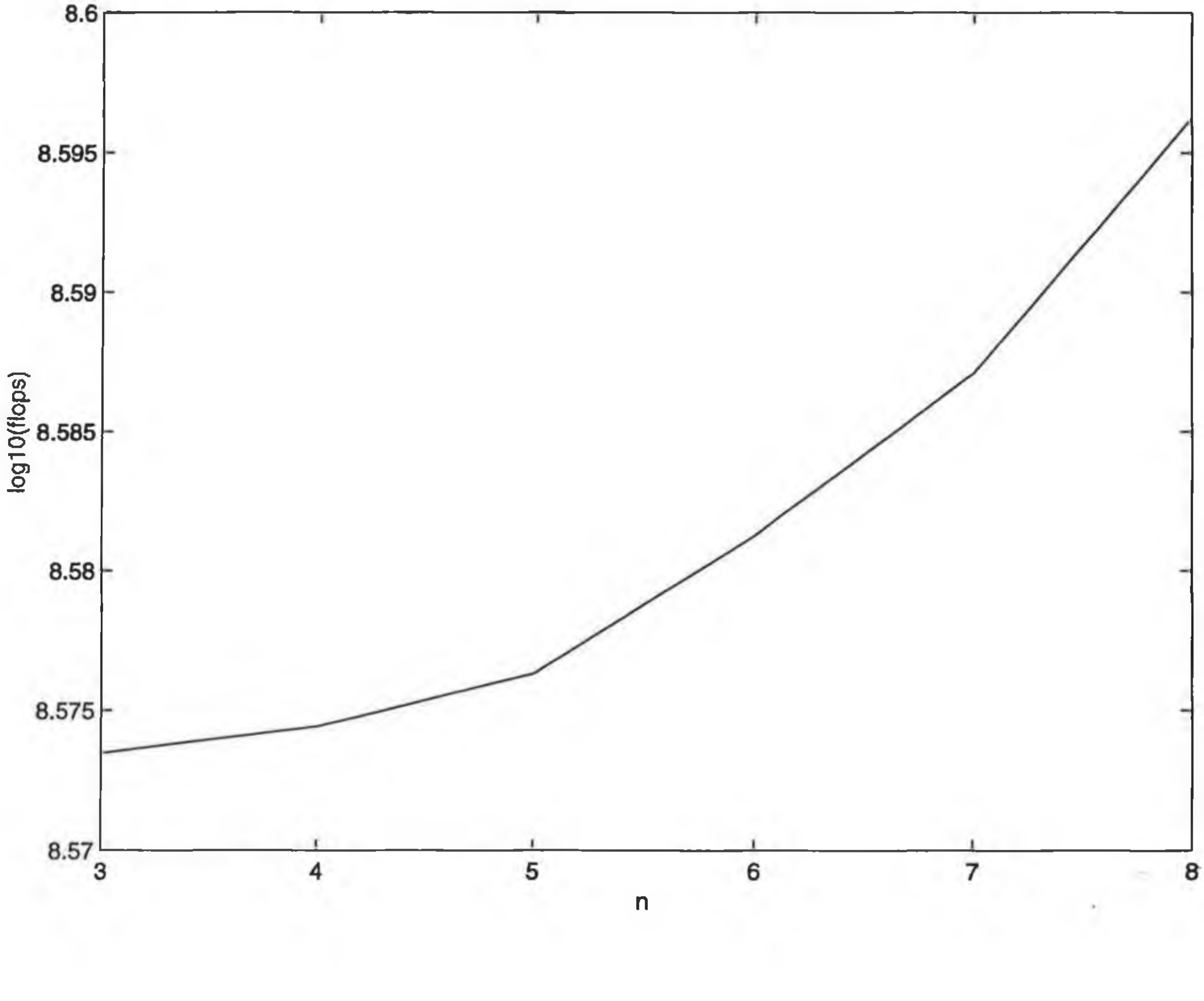

Figure 5.9: $log_{10}(flops)$ vs. n, $\epsilon = 0.001$ for Two Disc example

Case of n = 3				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	395713619
0.001	0.014803	0.014803	12	396227045
0.0001	0.014803	0.014803	12	396740471
1e-005	0.014803	0.014803	12	397253897

Case of n = 4				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	397767323
0.001	0.014803	0.014803	12	398280749
0.0001	0.014803	0.014803	12	398794175
1e-005	0.014803	0.014803	12	399307601

Case of n = 5				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	399821027
0.001	0.014803	0.014803	12	400334453
0.0001	0.014803	0.014803	12	400847879
1e-005	0.014803	0.014803	12	401361305

Case of n = 6				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	401874731
0.001	0.014803	0.014803	12	402388157
0.0001	0.014803	0.014803	12	402901583
1e-005	0.014803	0.014803	12	403415009

Case of n = 7				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	403928435
0.001	0.014803	0.014803	12	404441861
0.0001	0.014803	0.014803	12	404955287
1e-005	0.014803	0.014803	12	405468713

Case of n = 8				
ϵ	μ	λ	iterations	flops
0.01	0.014803	0.014803	12	405982139
0.001	0.014803	0.014803	12	406495565
0.0001	0.014803	0.014803	12	407008991
1e-005	0.014803	0.014803	12	407522417

Table 5.3: Results for the time domain problem applied to a servo motor.

The algorithm was applied to the specification

$$|||W_1 S|||_\infty \leq \lambda$$

For this problem there are known analytical solutions. This is useful for the purposes of software validation. With

$$G = \frac{-s+3}{-s+4}$$

and

$$W_1 = \frac{s+12}{s+1}$$

the optimal solution is 26.25. The results obtained are given in Table 5.4. A plot of this data is given in Figure 5.13.

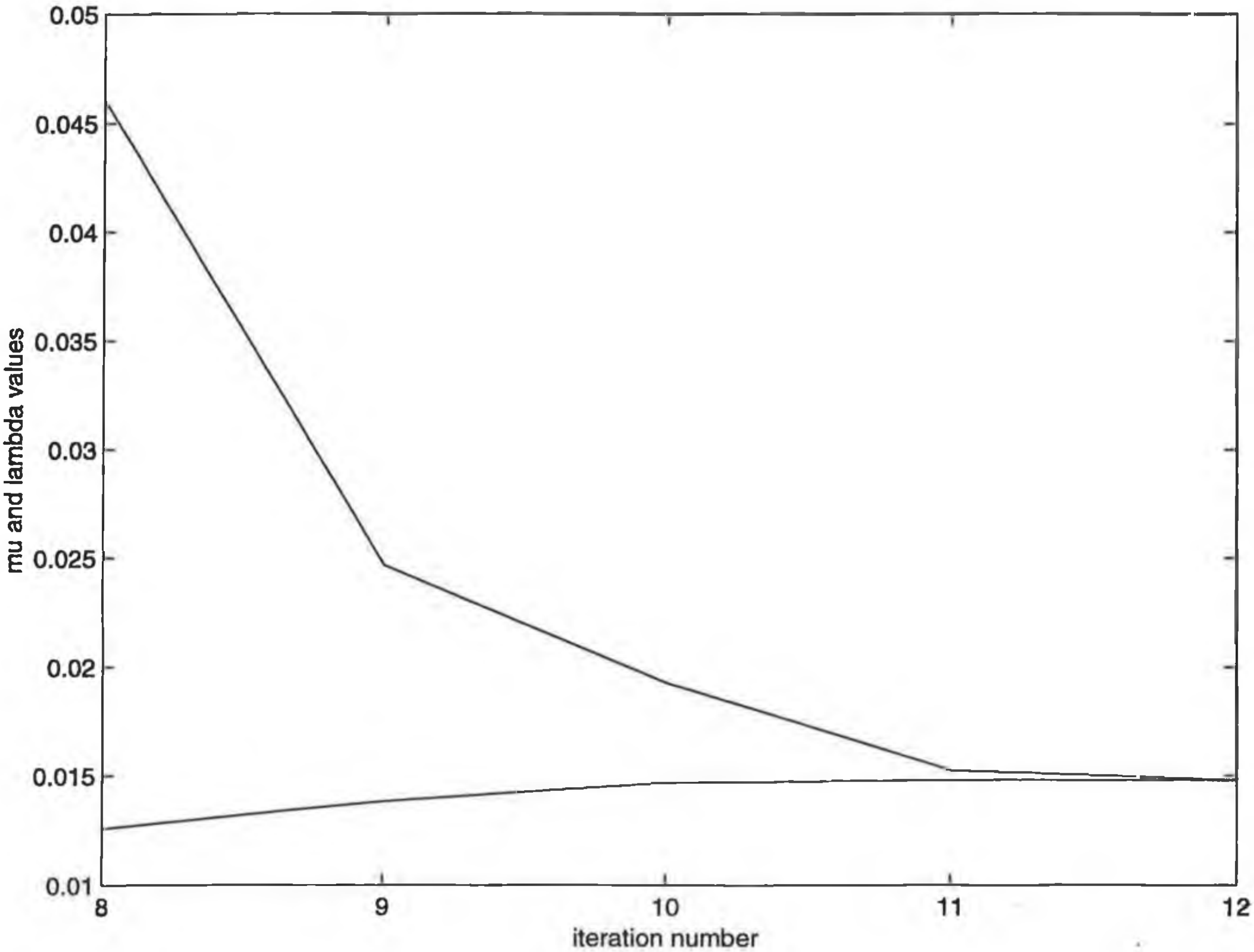

Figure 5.10: μ and λ values vs. iteration number for $\epsilon = 0.001$ for time response example

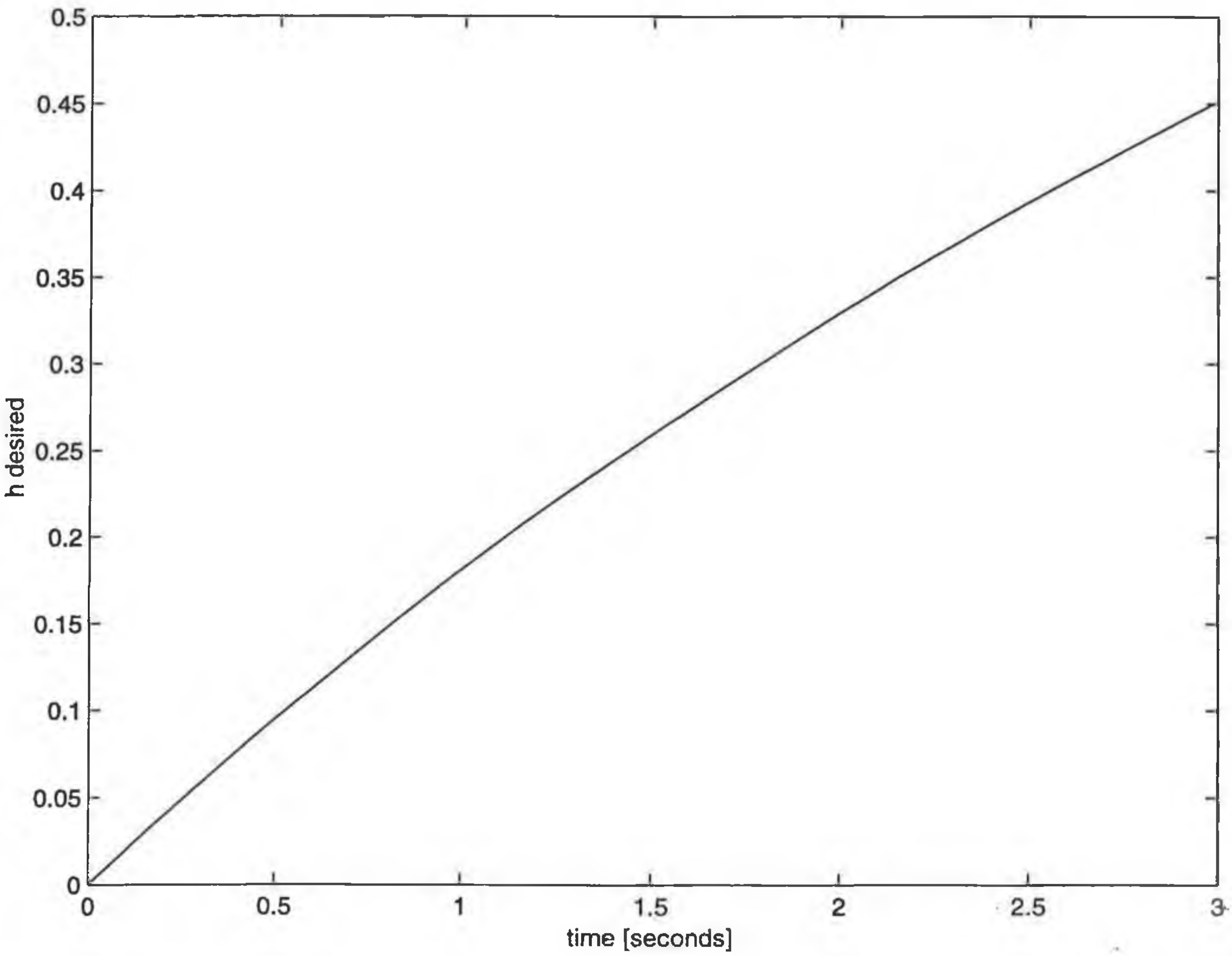

Figure 5.11: Desired step response for time response example

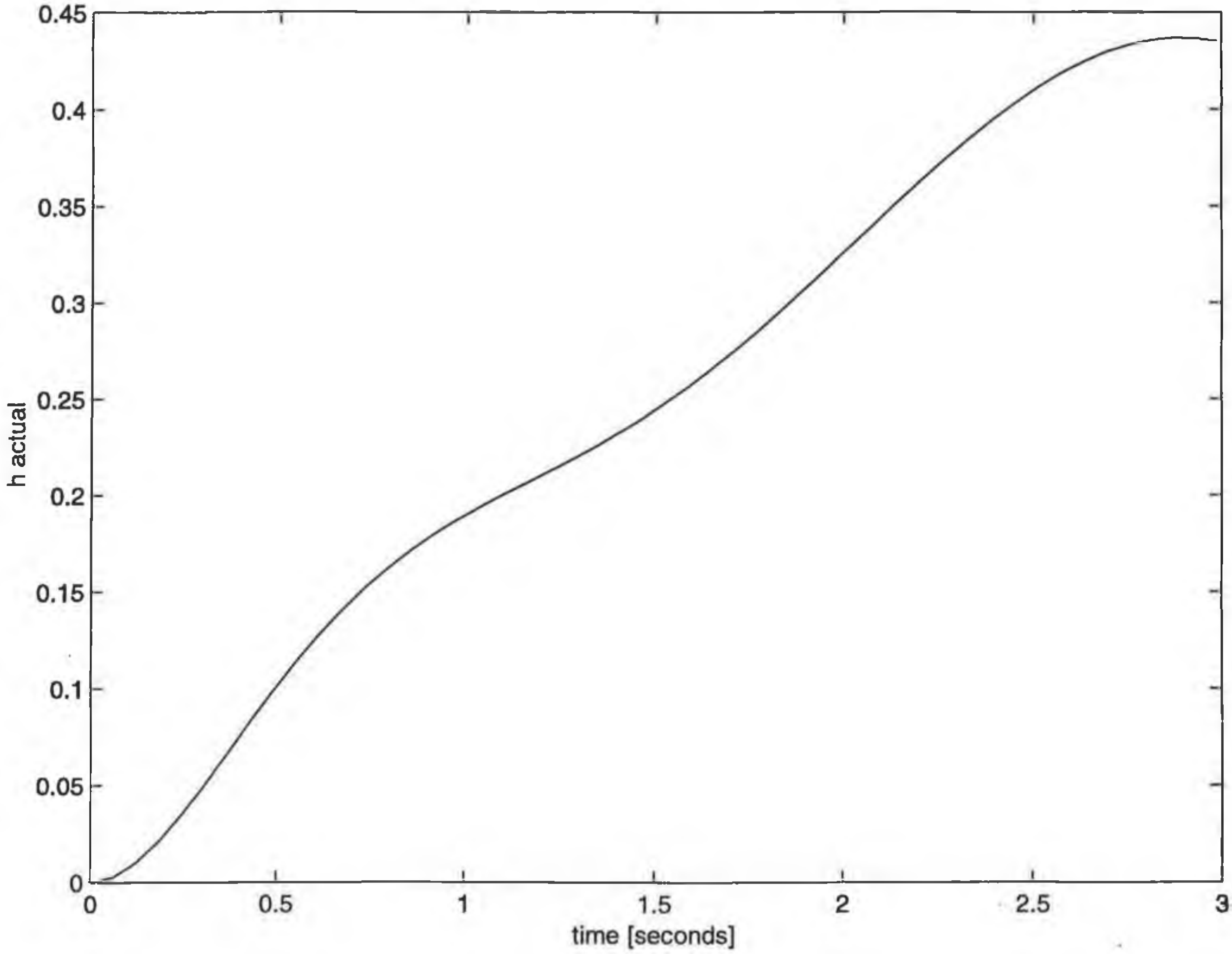

Figure 5.12: Actual step response for time response example

Case of n = 3				
ϵ	μ	λ	iterations	flops
2	41.6243	39.7263	8	6371788
1.5	40.4912	40.0502	9	6617334
1	40.4912	40.0502	9	6862876
0.5	40.4912	40.0502	9	7108422

Case of n = 4				
ϵ	μ	λ	iterations	flops
2	35.9766	35.1194	12	7833520
1.5	35.9766	35.1194	12	8558622
1	35.9766	35.1194	12	9283720
0.5	35.5709	35.3187	14	10584521

Table 5.4: Results for the weighted sensitivity function specification.

5.3 Software Validation

From the results obtained for the various examples given in the previous section it will be shown here that the coded algorithm did in fact display properties given in the previous chapter.

Convergence verification

- Looking at the plots of the tabulated data it can clearly be seen that the λ_i's converge to the optimal value of μ from below. This is shown for various example problems as seen in Figures 5.1, 5.6, 5.10 and 5.13. It can also been seen from these plots that the λ_i's form a non-decreasing monotonic sequence.

- It can also be seen from these plots that the μ_i's converge to μ from above. This convergence to the optimal answer is not necessarily monotonic. In the time response example (Figure 5.10) and the $||WS||_\infty$ example (Figure 5.13) μ_i decreases with each iteration, but in the robust disturbance attenuation problem (Figure 5.1) and the two-disc problem (Figure 5.6) μ_i is not monotonic decreasing.

- In all the examples with plots of the λ_i's and μ_i's against iteration number it can be seen that the algorithm converges, as required. At no stage do they cross over each other. So at all stages $\lambda_i \leq \mu_i$, and equality would hold if and only if the exact optimal solution is obtained.

Speed of convergence

- It must be said that from a practical point of view the algorithm did in fact converge very quickly. In fact, in all the examples given, convergence needed less than two minutes. It can be seen from the tables that the number of flops was increasing, but still a very short time was required.

Increasing n

- It can be seen from the example results in the tables and plots that the algorithm behaves as expected for varying n.

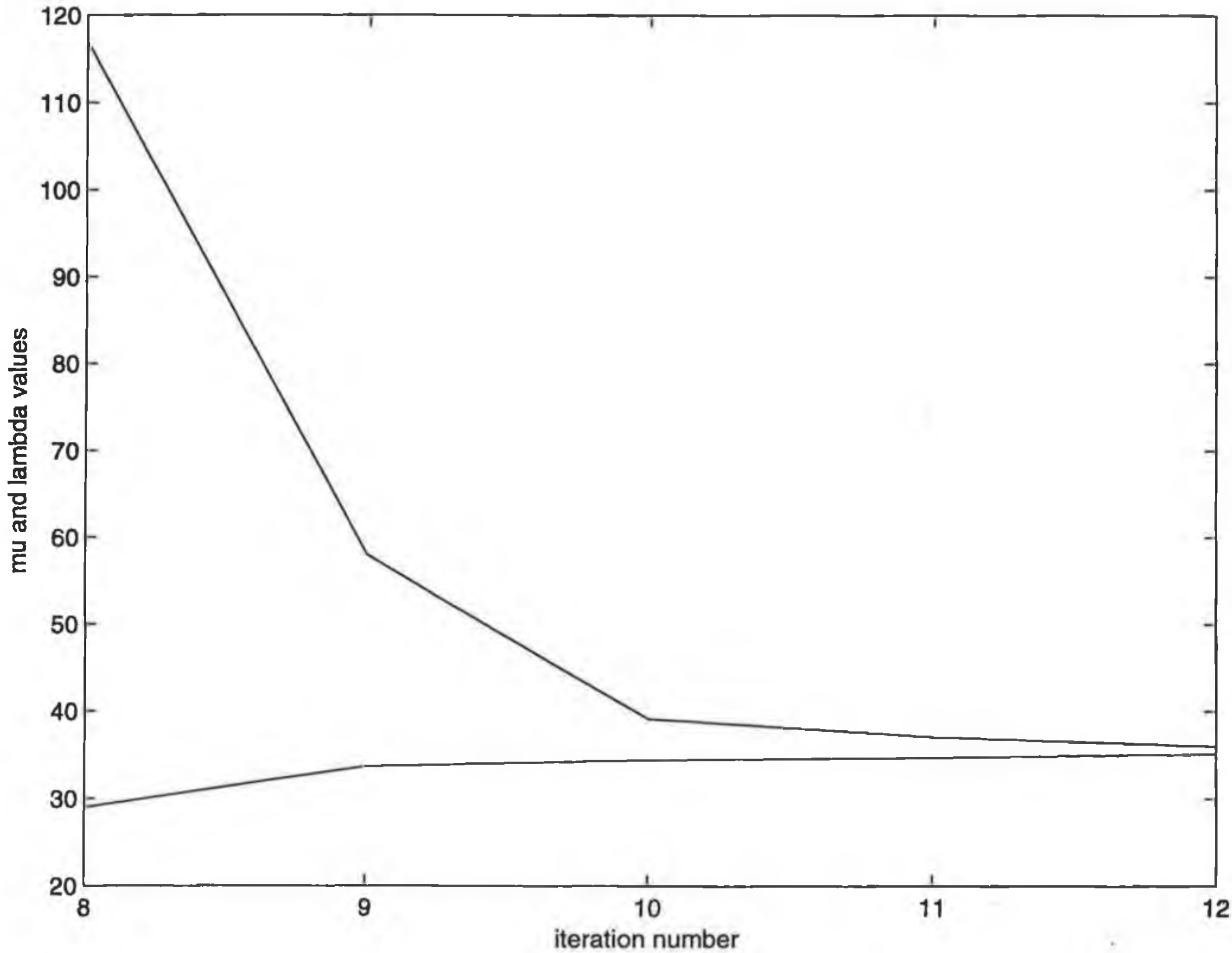

Figure 5.13: μ and λ values vs. iteration number for $\epsilon = 1$ and $n = 4$ for $\|WS\|_\infty$ example with known analytical solution

- Looking at the robust disturbance attenuation problem (Figure 5.2) and the two-disc problem (Figure 5.7) it is clear that for different values of n the bounds do not contradict. If for a particular value of n a range of values is found which contains the best approximation solution then if we increase n we should find a range which is no further from the optimal solution. The interval in which μ lies may have a larger upper limit for a larger n, but this is not in contradiction with the algorithm provided there is still a range in which μ may lie and not do worse for a larger n.

Analytical example

- The algorithm was applied to the specification

$$\|W_1 S\|_\infty \leq \lambda$$

and the results are given in Table 4. This is an example with a known analytical solution.

- It can clearly be seen from the table values for increasing n that the algorithm is moving closer to the known optimum of 26.25. By increasing the number of terms n in the expression for m the obtained solution is moving closer to the known solution.

- It can also be seen from the table that by decreasing the allowed difference ϵ for termination in step 4 (and hence possibly increasing the number of iterations before reaching the termination condition) it is seen that the solution approaches the known analytical solution.

- Although not shown here in the table it was seen that by an effective choice of poles for $m \in M$, the algorithm achieved a value close to the optimal solution. By an effective choice it is meant that the poles are selected at the locations of the known solution. Any alternative choice of poles did not give an improved result.

5.4 Another Example

This additional RDAP example has been included in order to demonstrate that reasonable controllers are obtained from the software.

Specifically, consider finding the controller which minimizes

$$\inf \| \; |W_1 S| + |W_2 T| \; \|_\infty$$

where, as usual,

$$S = \frac{1}{1+L}, \qquad T = \frac{L}{1+L}$$

L is the loop gain, and W_1 and W_2 are weighting functions. The minimization is over all feedback controllers which stabilize the closed loop system. This problem is again an instance of the robust disturbance attenuation problem (RDAP). The plant is

$$G = \frac{1}{s^2}$$

and the weights are

$$W_1 = \frac{(s+10)^2}{100(s+1)^2}, \qquad W_2 = \frac{(10s+1)^2}{100(s+1)^2}$$

Applying the algorithm to it yields the controller

$$K = \frac{0.5911s^3 + 0.5556s^2 + 0.5734s + 0.1147}{0.1147s^3 + 0.5734s^2 + 0.5556s + 0.5911}$$

$$= 5.1549\frac{(s+0.3489+0.8242i)(s+0.3489-0.8242i)(s+0.2422)}{(s+0.4355+1.0290i)(s+0.4355-1.0290i)(s+4.1289)}$$

The corresponding $|S(j\omega)|$ and $|T(j\omega)|$ are shown in Figure 5.14, while the loop gain's Bode diagrams are shown in Figure 5.15. From a robust control perspective, this is a good design. It is an intriguing design from a classical loop shaping perspective.

5.5 Computational Experience

The following are some items of advice on how to implement a coding of the algorithm, derived from experience.

Choice of α's

The first item is a likely pitfall. At each iteration the algorithm finds a new set of α's. The α's are obtained from the solution of the linear constraints (except for the first iteration in which case the α's used are those obtained from the initialisation step). The α's are used in the calculation of both λ_p and μ_p. The λ_p and μ_p values for each iteration are the upper and lower bounds which converge to the optimal solution and which are used to provide termination of the algorithm. The program is terminated once the upper and lower bounds are within a preset distance of each other. Once this condition has been met a solution has been obtained. The problem lies in the non-monotonic nature of the μ_p values. While a subsequence of the μ_p values converges to the optimal answer, μ, the final μ_p which was used in termination may not have been the minimum μ_p to date. The minimum μ_p to date was obtained from the α's which gave,

$$\min_p \|m_p - x_o\|$$

Therefore it is the α's which gave the lowest μ_p rather than the most recent μ_p which should be used in the solution.

Frequency response

It is most effective to compute the frequency response of each x_i once at the start and then sum responses to evaluate m rather than evaluate the frequency response of the new m_{i+1} due to the newly obtained α's. This would require evaluating a new frequency response for each iteration.

Linear program solvers

The linear program solvers tried were written as separate functions which just required the matrices of constraints and objective function as input. This was effective as it meant the solver was not problem specific and could be used for all

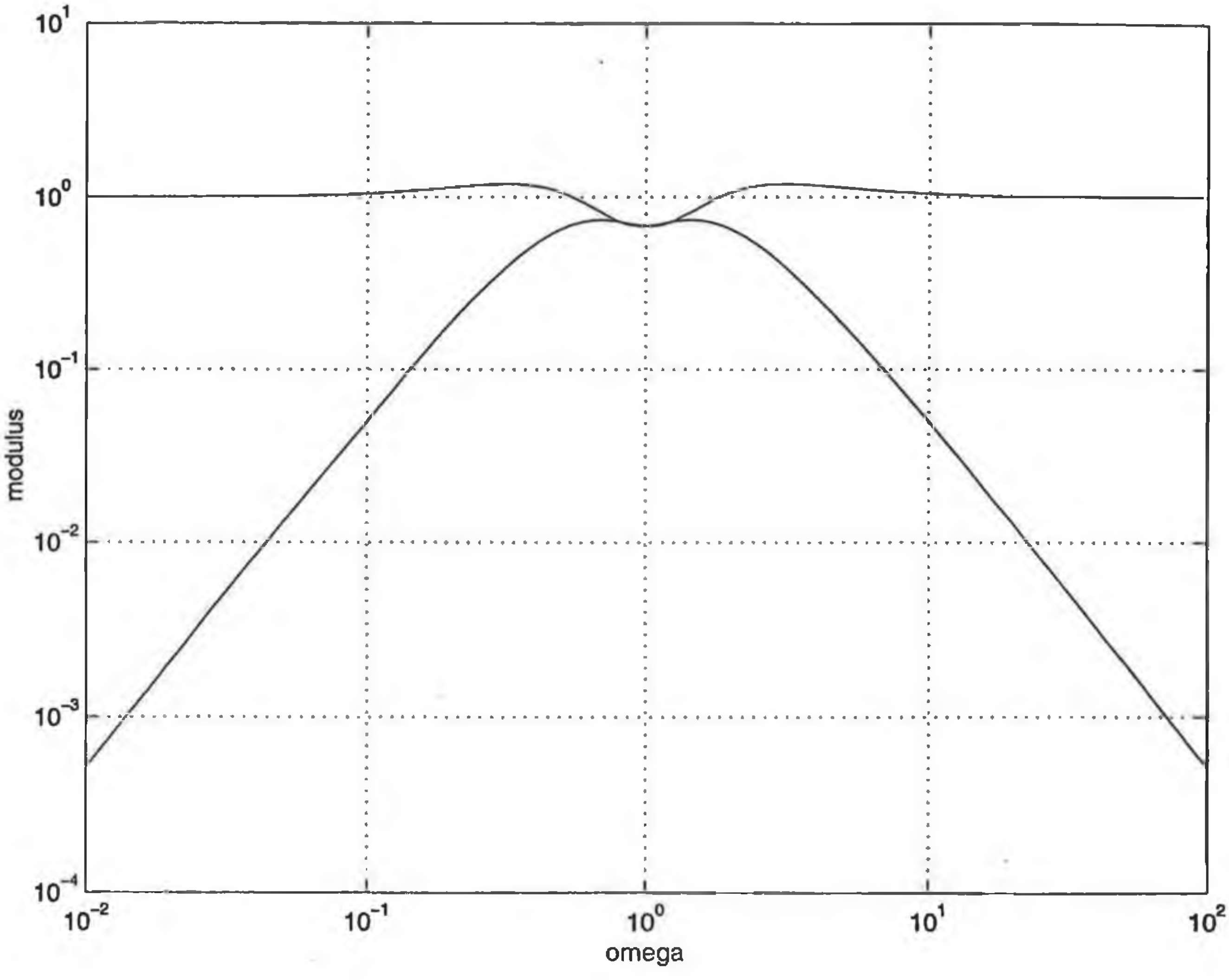

Figure 5.14: $|S|$ and $|T|$ for second RDAP problem

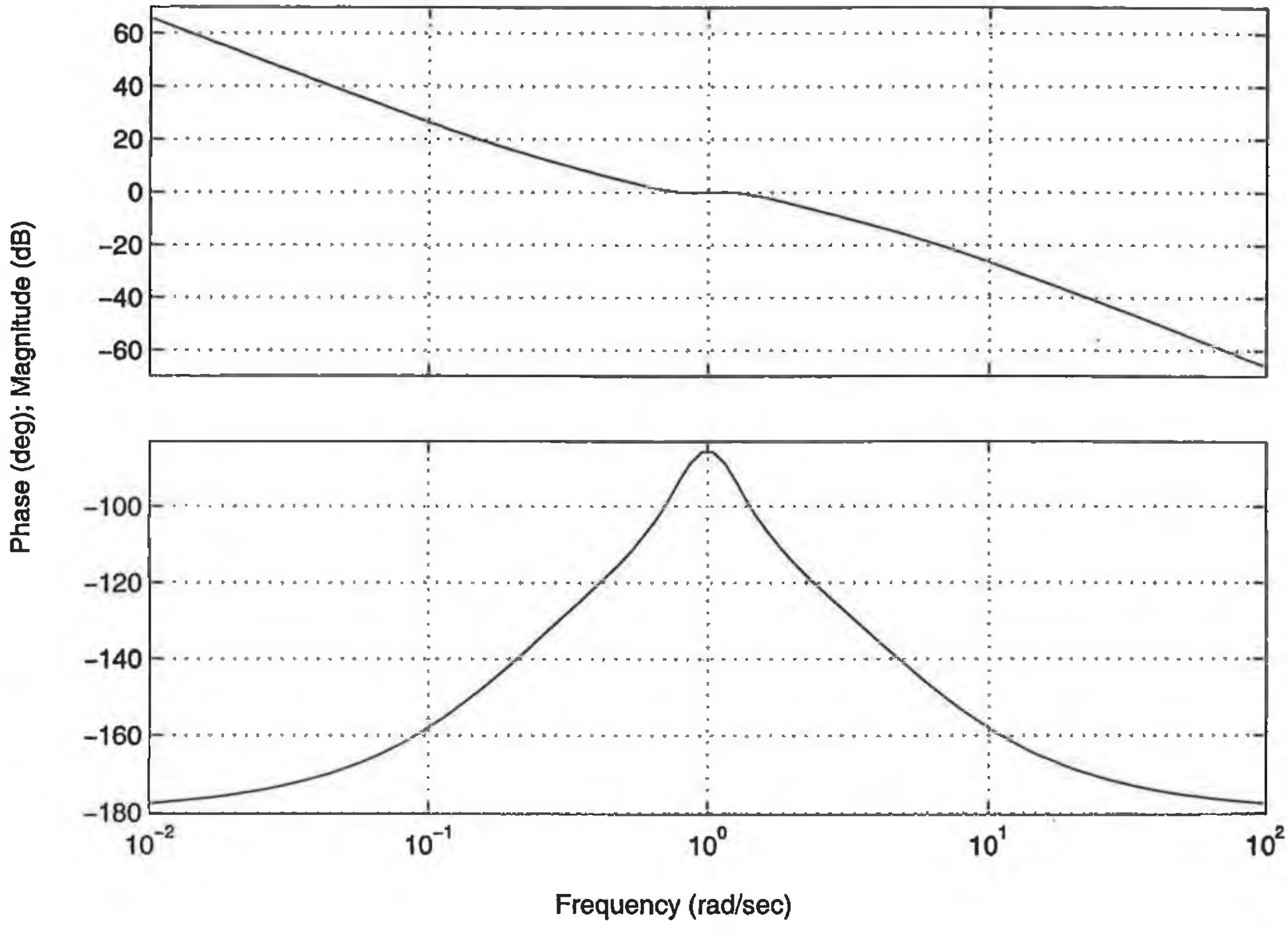

Figure 5.15: Loop gain's Bode diagram for second RDAP problem

the examples. It also meant that it was possible to use different linear program solvers and compare their computing times.

5.6 Discussion

The reasons for selecting the Akilov and Rubinov algorithm were outlined. Through an examination of the algorithm's performance the faith in the procedure was justified. The software validation of the paradigm using the Akilov and Rubinov algorithm obeyed all the properties expected and clearly demonstrated the advantages of this approach. Some computational experience was given as advice. It has been shown in this chapter that this is a very useful and practical algorithm and that it is a good choice for the purposes of this project.

Chapter 6

Duality Theory of Linear Programming

Given a particular optimization problem, it is generally possible to associate another optimization problem with it. The first and original problem is called the primal, while the second and derivative problem is called the dual problem. The two problems are intimately related. Duality techniques are a powerful research tool in the theory of optimization. This chapter develops the duality theory of linear programming.

6.1 Relationships between Dual and Primal Problems

As is well known, the situation with finite linear programs is as follows. The standard (finite) linear programming problem may be written as

$$\min_{x} c^T x \qquad \text{subject to}$$

$$Ax = b, \quad \text{and} \quad x \geq 0$$

This problem is known as the primal problem. Suppose that A is $m \times n$. It is formulated in the vector space $V = R^n$, so $x \in R^n$.

The corresponding problem in the dual space $V^* = R^m$ is known as the dual problem. For the primal problem above the dual problem is the following, where z is a vector in the dual space $z \in R^m$,

$$\max_z z^T b \qquad \text{subject to}$$

$$z^T A \leq c^T$$

Next, the relationship between the above two problems is developed.

Property 1 The value of the objective function in the primal problem is greater than or equal to the value in the dual problem, provided the x and z vectors in their respective spaces are feasible.

Proof:

The objective function for the dual problem is given by $z^T b$. If x is a feasible solution then $Ax = b$ and so

$$z^T b = z^T Ax \tag{6.1}$$

If z is a feasible solution then $z^T A \leq c^T$. Since $x \geq 0$ multiplication by x does not change the sense of the inequalities, so that

$$z^T Ax \leq c^T x$$

Combining gives

$$z^T b \leq c^T x \tag{6.2}$$

as claimed □

This property applies to the infinite case.

Property 2 The primal problem has an optimal solution if and only if the dual problem has an optimal solution. The objective functions in the primal and dual problems then have the same optimal value.

Proof:

The primal problem is given by the equations

$$Ax = b, \qquad x \geq 0,$$

Define

$$z_x = c^T x$$

This can be rewritten as

$$c^T x - z_x = 0$$

This provides the starting point for the Simplex Method

$$\begin{pmatrix} A & 0 \\ c_T & -1 \end{pmatrix} \begin{pmatrix} x \\ z_x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} \tag{6.3}$$

Since the original problem had n variables and m constraints and if the basic variables are the first m variables of the x vector then the A matrix can be rewritten as

$$A = \begin{pmatrix} A_{basic} & A_{nonbasic} \end{pmatrix}$$

Write

$$x = \begin{pmatrix} x_{basic} \\ x_{nonbasic} \end{pmatrix}$$

Then $x_{nonbasic} = 0$ for a basic solution.

Since A_{basic} is an $m \times m$ full rank matrix the solution set of $Ax = b$ can be written as

$$Ax = b = \begin{pmatrix} A_{basic} & A_{nonbasic} \end{pmatrix} \begin{pmatrix} x_{basic} \\ x_{nonbasic} \end{pmatrix} = A_{basic}x_{basic} + A_{nonbasic}x_{nonbasic}$$

$$\implies x_{basic} + A_{basic}^{-1}A_{nonbasic}x_{nonbasic} = A_{basic}^{-1}b$$

$$\implies x_{basic} = A_{basic}^{-1}b - A_{basic}^{-1}A_{nonbasic}x_{nonbasic}$$

Also

$$z_x = c^T x = c_{basic}^T x_{basic} + c_{nonbasic}^T x_{nonbasic}$$

and substituting for x_{basic} gives

$$z_x = c_{basic}^T \left(A_{basic}^{-1} b - A_{basic}^{-1} A_{nonbasic} x_{nonbasic} \right) + c_{nonbasic}^T x_{nonbasic}$$

$$= c_{basic}^T A_{basic}^{-1} b - c_{basic}^T A_{basic}^{-1} A_{nonbasic} x_{nonbasic} + c_{nonbasic}^T x_{nonbasic}$$

Hence,

$$z_x = z_{basic} + r_{basic}^T x_{nonbasic}$$

where

$$z_{basic} = c_{basic}^T A_{basic}^{-1} b$$

is the current objective function value, and

$$r_{basic}^T = c_{nonbasic}^T - c_{basic}^T A_{basic}^{-1} A_{nonbasic}$$

is the relative cost vector.

Therefore (6.3) can be reformatted as follows

$$\begin{pmatrix} A & 0 \\ c^T & -1 \end{pmatrix} \begin{pmatrix} x \\ z_x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} A_{basic} & A_{nonbasic} & 0 \\ c_{basic}^T & c_{nonbasic}^T & -1 \end{pmatrix} \begin{pmatrix} x_{basic} \\ x_{nonbasic} \\ z_x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

Multiplying on the left by $\begin{pmatrix} A_{basic}^{-1} & 0 \\ 0 & 1 \end{pmatrix}$ gives

$$\begin{pmatrix} I & A_{basic}^{-1} A_{nonbasic} & 0 \\ c_{basic}^T & c_{nonbasic}^T & -1 \end{pmatrix} \begin{pmatrix} x_{basic} \\ x_{nonbasic} \\ z_x \end{pmatrix} = \begin{pmatrix} A_{basic}^{-1} b \\ 0 \end{pmatrix}$$

Multiplying on the left by $\begin{pmatrix} -c^T_{basic} & 1 \end{pmatrix}$, and using the result as the last row, gives

$$\begin{pmatrix} I & A^{-1}_{basic}A_{nonbasic} & 0 \\ 0 & c^T_{nonbasic} - c^T_{basic}A^{-1}_{basic}A_{nonbasic} & -1 \end{pmatrix} \begin{pmatrix} x_{basic} \\ x_{nonbasic} \\ z_x \end{pmatrix} = \begin{pmatrix} A^{-1}_{basic}b \\ -z_{basic} \end{pmatrix} \qquad (6.4)$$

With the problem formatted as in (6.4) the relative cost vector is given by

$$r^T_{basic} = c^T_{nonbasic} - c^T_{basic}A^{-1}_{basic}A_{nonbasic}$$

If the present basic feasible solution is optimal, then

$$r^T_{basic} \geq 0$$

Hence,

$$c^T_{nonbasic} - c^T_{basic}A^{-1}_{basic}A_{nonbasic} \geq 0$$

giving,

$$c^T_{nonbasic} \geq c^T_{basic}A^{-1}_{basic}A_{nonbasic} \qquad (6.5)$$

Let z be defined by

$$z^T = c^T_{basic}A^{-1}_{basic}$$

Then

$$z^T A = \begin{pmatrix} z^T A_{basic} & z^T A_{nonbasic} \end{pmatrix}$$

so

$$z^T A = \begin{pmatrix} c^T_{basic} & c^T_{basic}A^{-1}_{basic}A_{nonbasic} \end{pmatrix}$$

and from (6.5)

$$z^T A \leq \begin{pmatrix} c^T_{basic} & c^T_{nonbasic} \end{pmatrix} = c^T$$

so

$$z^T A \leq c^T$$

and hence z is a feasible solution of the dual problem. This shows that if the primal problem has a finite optimal solution then the dual problem is feasible. Since

$$z^T = c_{basic}^T A_{basic}^{-1}$$

the following is also true,

$$z^T b = c_{basic}^T A_{basic}^{-1} b$$

and using the fact that $x_{nonbasic} = 0$ for a basic solution, $A_{basic}^{-1} b$ can be replaced by x_{basic}, giving

$$z^T b = c_{basic}^T x_{basic} = c^T x$$

Combining this with property 1 shows that the primal and dual optimal costs are equal, and therefore this z is the optimal solution of the dual problem.

This shows that given an optimal solution, x, of the primal problem, an optimal solution z of the dual problem exists and the values of the objective functions are equal. The fact that an optimal x exists given an optimal z follows by viewing the dual problem as a primal problem. □

This property applies to the finite case.

Property 3 If either the primal or dual problem has an unbounded optimal solution then the other problem has an empty feasible solution set.

Proof:

Suppose that the primal problem has unbounded feasible solutions and the dual problem has a feasible solution, $z_{feasible}$. The dual objective function is $z_{feasible}^T b$. Since the primal problem has unbounded solutions it is possible to find a feasible x which gives $c^T x \leq z_{feasible}^T b$. This contradicts Property 1 and so if the primal problem has unbounded feasible solutions then the dual feasible set is empty.

The same arguement could be made if the dual problem has unbounded feasible solutions and the primal problem has a feasible solution, $x_{feasible}$. So if the dual problem has unbounded feasible solutions then the primal feasible set is empty. □

Property 1 is very useful because any feasible solution of the dual problem is a lower bound on the primal solution. Properties 1 and 3 hold in the infinite and semi-infinite cases. However, Property 2 holds for the finite case only. Thus the duality theory for semi-infinite and infinite linear programs is more subtle than the finite case.

6.2 Formatting in Standard Form

In the context of this thesis, the convex specifications provide inequality constraints involving sign free variables. Such linear programs are a little different from the formulation discussed above. In such a case the primal linear programming problem is given by

$$\min_x c^T x \qquad \text{subject to}$$

$$Ax \leq b, \quad x_i\text{'s are sign free variables}$$

This problem may be written equivalently as

$$\min_x \begin{pmatrix} c^T & -c^T & 0 \end{pmatrix} \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix}$$

subject to the constraints

$$\begin{pmatrix} A & -A & I \end{pmatrix} \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix} = b, \quad x_+ \geq 0, \; x_- \geq 0, \; y \geq 0$$

Letting,

$$\hat{c} = \begin{pmatrix} c^T & -c^T & 0 \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix}, \quad \hat{A} = \begin{pmatrix} A & -A & I \end{pmatrix}, \quad \hat{x} \geq 0$$

The problem is now in the form,

$$\min_x \hat{c}^T \hat{x} \qquad \text{subject to}$$

$$\hat{A}\hat{x} = b, \text{ and } \hat{x} \geq 0$$

This is the format discussed earlier in this chapter. For the primal problem above the dual problem is then the following, where z is a vector in the dual space

$$\max_z z^T b \qquad \text{subject to}$$

$$\hat{A}^T z \leq \hat{c}$$

which when written fully becomes,

$$\max_z z^T b$$

subject to the constraints

$$\begin{pmatrix} A^T \\ -A^T \\ I \end{pmatrix} z \leq \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}$$

Writing the constraints out fully gives,

$$A^T z \leq c,$$

$$-A^T z \leq -c,$$

$$Iz \leq 0,$$

These three constraints can be reduced to the following two constraints,

$$A^T z = c,$$

$$z \leq 0,$$

So now we have the dual of our original problem with the inequality constraints and the sign free variables x. The dual problem is

$$\max_z z^T b \qquad \text{subject to}$$

$$A^T z = c, \text{ and } z \leq 0$$

6.3 Bounds on Semi-infinite Linear Programs

This section develops some bounds for the semi-infinite linear programming problems [12].

Consider an infinite dimensional linear program of the form

$$\lambda_t^p = \min c^T x \qquad \text{subject to}$$

$$Ax \leq b, \ x_i\text{'s are sign free variables,}$$

Here A will be an $\infty \times \infty$ matrix and x will be $\infty \times 1$. Let the optimal cost, if it exists, be λ_t^p, where p stands for "primal" and t stands for "true".

Suppose that this infinite primal problem is truncated so that only finitely many variables (FMV) are taken. The problem then becomes a semi-infinite problem. In this case A will be a $\infty \times n$ matrix and x will be $n \times 1$. Let the optimal cost, if it exists, be denoted by λ_v^p. If this problem can be solved then its optimal solution x_v is feasible for the optimal infinite problem and λ_v^p is clearly an upper bound for λ_t^p.

$$\lambda_v^p \geq \lambda_t^p \tag{6.6}$$

Now suppose that the infinite problem is truncated by taking only finitely many constraints (FMC), but all infinitely many variables are retained. Again the problem becomes a semi-infinite problem, but in this case A will be a $n \times \infty$ matrix and x will be $\infty \times 1$. Let the optimal cost, if it exists, be denoted by λ_c^p. If this problem can be solved λ_c^p is clearly a lower bound for λ_t^p, giving

$$\lambda_c^p \leq \lambda_t^p \tag{6.7}$$

In this case, the FMC optimal solution x_c may or may not be feasible for the original infinite problem.

If the solutions exist combining (6.6) and (6.7) gives

$$\lambda_c^p \leq \lambda_t^p \leq \lambda_v^p \tag{6.8}$$

As the number of variables and constraints retained increases λ_c^p and λ_v^p may move closer to each other. For practical purposes they may or may not get to be close enough to the optimal λ_t^p.

Consider the FMV semi-infinite problem as described above. In such a case A will be a $\infty \times n$ matrix, x will be $n \times 1$ and b will be a $\infty \times 1$ matrix. Recall that the optimal cost is denoted by λ_v^p.

Suppose that the number of constraints in this semi-infinite primal problem are truncated. The problem is now a finite problem. If the optimal cost of this finite problem is denoted by λ_{vc}^p then using the idea as in the FMC case above $\lambda_{vc}^p \leq \lambda_v^p$. Therefore the finite case provides a lower bound on the semi-infinite case. However, x_{vc} may not be feasible.

As shown in the previous section the dual problem of the finite primal problem above is given by

$$\max_z z^T b \qquad \text{subject to}$$

$$A^T z = c, \quad z \leq 0$$

with A^T being $n \times m$, z being $m \times 1$ and c being $n \times 1$.

Suppose that the formal dual of the above infinite primal problem has an optimal solution. Denote it by λ_t^d, with the subscript d to denote "dual ". In this case rows in

A^T correspond to variables in the primal and columns in A^T correspond to constraints in the primal. Note that the dual problem is a maximization rather a minimization problem.

Again suppose that this infinite dual problem is truncated so that only finitely many variables (FMV) are taken. The problem then becomes a semi-infinite problem. Let the optimal cost, if it exists, be denoted by λ_v^d. If this problem can be solved then its optimal solution z_v is feasible for the optimal infinite problem and λ_v^d is a lower bound for λ_t^d.

$$\lambda_v^d \leq \lambda_t^d \tag{6.9}$$

Now suppose that the infinite dual problem is again truncated, but now only finitely many constraints (FMC) are taken, but all infinitely many variables are retained. Again the problem becomes a semi-infinite problem. Let the optimal cost, if it exists, be denoted by λ_c^d. If this problem can be solved λ_c^d is an upper bound for λ_t^d, giving

$$\lambda_t^d \leq \lambda_c^d \tag{6.10}$$

In this case, the FMC optimal solution x_c may or may not be feasible for the original infinite problem.

If the solutions exist combining (6.9) and (6.10) gives

$$\lambda_v^d \leq \lambda_t^d \leq \lambda_c^d \tag{6.11}$$

So looking at the infinite problem and finitely many variables semi-infinite problems there are two inequalities for the solutions to these problems

$$\lambda_v^p \geq \lambda_t^p$$

and

$$\lambda_v^d \leq \lambda_t^d$$

If there was no duality gap (i.e. the solutions to the primal and dual infinite problems were equal) then inequalities could be combined giving,

$$\lambda_v^p \geq \lambda_t^p = \lambda_t^d \geq \lambda_v^d \tag{6.12}$$

It is worth noting that if there is a duality gap and the primal and dual infinite problems are not equal then Property 1 provides an inequality. This property applies to the infinite problems. It must first be noted that the variables and constraints for the linear program obtained from the Akilov and Rubinov algorithm are the form of the dual problem as stated at the start of this chapter. The variables are sign free and the constraints are inequalities. The property states that the value of the objective function in the primal problem is greater than or equal to the value in the dual problem, provided the x and z vectors in their respective spaces are feasible. But taking that the fact that the primal problem given in this section has it's inequalities and variables in the dual form of this property the following inequality holds if there is a duality gap

$$\lambda_t^p \leq \lambda_t^d \tag{6.13}$$

The Akilov and Rubinov algorithm can be used to get solutions for λ_v^p and as such gives an upper bound on the true infinite case optimal λ_t^p. It would be useful to have a method of solving the semi-infinite finitely many variables dual problem to obtain a value for λ_v^d. The true optimal would be between these values and so an effective evaluation could be made of the cost of the H^∞ truncation used in proposed algorithm. There are a number of possible difficulties with this approach for evaluating the issue of H^∞ truncation. Unfortunately the semi-infinite finitely many variables dual problem is very complicated. There are difficulties arising from the fact that the semi-infinite problem may not be feasible or bounded. If a solution does exist it will be convex but formulating it as a norm (for use in the Akilov and Rubinov algorithm) has further difficulties. The inequality (6.12) also requires that there is no duality gap which is not straightforward to show for all problems. A brief example is given in the next chapter.

Therefore a more sophisticated approach is required to determine the difference between the optimal λ_t^p and the Akilov and Rubinov obtained solution λ_v^p. This will be discussed in the next chapter.

6.4 An Example $\|W_1 S\|_\infty$

This section shows how to formulate the primal problem using a subspace of H_∞ to obtain a finitely many variables problem. The constraints are added for each iteration of the Akilov and Rubinov algorithm. The equivalent dual is then given.

This example uses the $\|W_1 S\|_\infty$ specification to give the problems involved, but it could be applied to any specification for which the algorithm is applicable.

As was shown in Chapter 3 the specification $\|W_1 S\|_\infty \leq \lambda$ can be reduced to a standard linear programming problem. The primal linear programming formulation gives inequality constraints and sign free variables, i.e.

$$\min_x c^T x$$

subject to

$$Ax \leq b$$

This is obtained by taking only finitely many constraints, where each constraint, say the i^{th}, corresponds to a certain ω and θ, namely (ω_i, θ_i). Thus,

$$\inf_{\alpha \epsilon R^n} \lambda \qquad \text{subject to}$$

$$\sum_{j=1}^{n} \alpha_j Re\left\{e^{j\theta_i} x_j(j\omega_i)\right\} \leq \lambda + Re\left\{e^{j\theta_i} x_0(j\omega_i)\right\} \qquad i = 1, 2, ..., N \tag{6.14}$$

Let $b_i = Re\left\{e^{j\theta_i} x_0(j\omega_i)\right\}$ and $a_{ij} = Re\left\{e^{j\theta_i} x_j(j\omega_i)\right\}$, so the above equation becomes

$$\sum_{j=1}^{n} \alpha_j a_{ij} \leq \lambda + b_i \qquad i = 1, 2, ..., N \tag{6.15}$$

or $< \alpha, a_i > \leq \lambda + b_i$ where $\alpha = (\alpha_1, \alpha_2, ...)^T$, $a_i = (a_{i1}, ..., a_{in})^T$. Now subtracting λ from both sides, gives

$$\sum_{j=1}^{n} \alpha_j a_{ij} - \lambda \leq +b_i \quad i = 1, 2, ..., N \tag{6.16}$$

This equation is of the primal linear programming format i.e. $Ax \leq b$. In the primal linear programming format with inequality constraints, i.e. $\min_x c^T x$ subject to $Ax \leq b$, the A matrix becomes,

$$A = \begin{pmatrix} Re\left\{e^{j\theta_1} x_1(j\omega_1)\right\} & Re\left\{e^{j\theta_1} x_2(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_1} x_n(j\omega_1)\right\} & -1 \\ \vdots & & & \vdots & \vdots \\ Re\left\{e^{j\theta_p} x_1(j\omega_p)\right\} & Re\left\{e^{j\theta_p} x_2(j\omega_p)\right\} & \dots & Re\left\{e^{j\theta_p} x_n(j\omega_p)\right\} & -1 \end{pmatrix} \tag{6.17}$$

where p is the number of constraints so far. The matrix A is a $p \times (n+1)$ matrix.

$$b = \begin{pmatrix} Re\left\{e^{j\theta_1} x_0(j\omega_1)\right\} \\ Re\left\{e^{j\theta_2} x_0(j\omega_2)\right\} \\ \vdots \\ Re\left\{e^{j\theta_p} x_0(j\omega_p)\right\} \end{pmatrix} \tag{6.18}$$

The vector b is a $p \times 1$ matrix.

$$x_T = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n & \lambda \end{pmatrix} \tag{6.19}$$

The vector x is a $(n+1) \times 1$ matrix.

$$c^T = \begin{pmatrix} 0 & \dots & 0 & 0 & 1 \end{pmatrix} \tag{6.20}$$

The vector c is a $(n+1) \times 1$ matrix.

So as worked out above the dual of this problem is

$$\max_z z^T b$$

subject to the constraints

$$A^T z = c,$$

$$z \leq 0,$$

with A, b and c as given above. So

$$A^T = \begin{pmatrix} Re\left\{e^{j\theta_1}x_1(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_1(j\omega_p)\right\} \\ Re\left\{e^{j\theta_1}x_2(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_2(j\omega_p)\right\} \\ Re\left\{e^{j\theta_1}x_n(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_n(j\omega_p)\right\} \\ -1 & \dots & -1 \end{pmatrix} \tag{6.21}$$

and b and c are given as in expressions (6.18) and (6.20) as before. Writing this dual problem in full form gives,

$$\max_z \begin{pmatrix} z_0 & z_1 & z_2 & \dots & z_p \end{pmatrix} \begin{pmatrix} Re\left\{e^{j\theta_1}x_0(j\omega_1)\right\} \\ Re\left\{e^{j\theta_2}x_0(j\omega_2)\right\} \\ \vdots \\ Re\left\{e^{j\theta_p}x_0(j\omega_p)\right\} \end{pmatrix}$$

subject to

$$\begin{pmatrix} Re\left\{e^{j\theta_1}x_1(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_1(j\omega_p)\right\} \\ Re\left\{e^{j\theta_1}x_2(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_2(j\omega_p)\right\} \\ \vdots & & \vdots \\ Re\left\{e^{j\theta_1}x_n(j\omega_1)\right\} & \dots & Re\left\{e^{j\theta_p}x_n(j\omega_p)\right\} \\ -1 & \dots & -1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$z \leq 0,$$

6.5 Conclusions

This chapter contained some relationships between primal and dual problems. These relationships were proved. Formulating the inequality contraints and variables from the convex problems in this thesis into primal and dual problems was shown. Bounds on the true optimal solution were obtained from the semi-infinite linear programs, but this lead to the need for an alternative approach to analyse the cost of truncation. This alternative approach will be outlined in the next chapter.

Chapter 7

Analysis of the Issue of Truncation

Functional analysis techniques are used to give a qualitative evaluation and a quantitative measurement of the cost of truncation. These results follow over to the Boyd-Barratt numerical approach.

7.1 H^∞ Duality Theory

This section describes the application of H^∞ duality theory to the $\|WS\|_\infty$ example problem [16,17,21,26].

We begin with some definitions [23]. Let X denote a normed linear space. Let M denote a finite dimensional normed linear subspace, with the norm induced by the norm on X. Let X^* be the set of bounded linear functionals on the vector space X, called the dual space of X. The set of all linear functionals $y \in X^*$ for which

$$< m, y >= 0 \;\; \forall \;\; m \in M$$

is called the annihilator of M, written $M^\perp$. The complex conjugates of the elements of M is denoted by $\overline{M}$. The prefix B before M denotes the set of elements of M with norm less than or equal to one, BM. The norm of a linear functional is defined to be

$$\|n\|_{X^*} = \sup_{x \in BX} | < x, n > |$$

It may be shown that

$$\|x\|_X = \sup_{n \in BX^*} | < x, n > |$$

Using this notation the primal problem can be written as follows

$$\inf_{m \in M} \| x_0 - m \|_X$$

In this case the equivalent dual problem is

$$\max_{n \in BM^\perp} | < x_0, n > |$$

where $M^\perp$ is a subspace of X^*. By the Hahn-Banach theorem

$$\inf_{m \in M} \| x_0 - m \|_X = \max_{n \in BM^\perp} | < x_0, n > | \tag{7.1}$$

The Hahn-Banach theorem also gives the relation

$$\min_{m \in M^\perp} \| x_0 - m \|_X = \sup_{n \in BM} | < x_0, n > | \tag{7.2}$$

Note that the existence of an element that achieves the optimum is assured in the dual space, but not necessarily in the primal space.

Consider the problem of finding the stabilizing controller which minimizes the weighted sensitivity function in the infinity-norm sense, namely

$$\inf_{Q \in H^\infty} \| WS \|_\infty$$

It is shown next that this problem has the structure of (7.1) and (7.2). As before by Youla parameterisation the following is obtained,

$$S = D(QN + V)$$

Substituting for S gives,

$$\inf_{Q \in H^\infty} \|WD(QN + V)\|_\infty = \inf_{Q \in H^\infty} \|WDQN + WDV\|_\infty \tag{7.3}$$

Note that any biproper transfer function can be separated into its inner and outer factors. This means a transfer function $F(s)$ can be written as

$$F = F_{op}F_{ip}$$

where op stands for "outer part" and ip stands for "inner part". Let,

$$F_{ip} = \prod \left(\frac{-s + z_i}{s + z_i} \right)$$

where z_i are all the right half plane (RHP) zeros of F. Then F_{ip} is called a Blaschke product. This gives

$$F = \left[\prod \left(\frac{-s + z_i}{s + z_i} \right) \right] F_{op}$$

Now by expressing N and D in terms of their inner and outer factorisations

$$N = N_{op}N_{ip}$$

$$D = D_{op}D_{ip}$$

and substituting into (7.3) gives,

$$\inf_{Q \in H^\infty} \|WD_{op}D_{ip}QN_{op}N_{ip} + WD_{op}D_{ip}V\|_\infty$$

It is easily verified that

$$|D_{ip}(j\omega)| = 1 = |N_{ip}(j\omega)| \;\; \forall \, \omega$$

So,

$$|WS(j\omega)| = |D_{ip}^{-1}N_{ip}^{-1}WS(j\omega)| \;\; \forall \, \omega$$

giving

$$\inf_{Q \in H^\infty} \|WS\|_\infty = \inf_{Q \in H^\infty} \left\| WD_{op}N_{op}Q + WD_{op}VN_{ip}^{-1} \right\|_\infty$$

Writing this problem in the form of the primal problem above gives,

$$\inf_{m\in M} \| x_0 - m \|_\infty = \min_{Q\in H^\infty} \| WD_{op}N_{op}Q + WD_{op}VN_{ip}^{-1} \|_\infty$$

With $X = C_0$, $x_0 = -WD_{op}VN_{ip}^{-1}$ and $M = A_0^\infty$ so that $X^* = NBV$ and $M^\perp = \overline{H^1}$ yields [16,17,21,26]

$$\inf_{m\in M} \| x_0 - m \|_\infty = \max_{n\in BM^\perp} | < x_0, n > |$$

$$= \inf_{m\in A_0^\infty} \| WD_{op}VN_{ip}^{-1} + m \|_\infty = \max_{n\in B\overline{H^1}} | < WD_{op}VN_{ip}^{-1}, n > |$$

$$= \max_{n\in B\overline{H^1}} \int WD_{op}VN_{ip}^{-1}\overline{n}\mathrm{d}\omega$$

$$= \max_{h\in BH^1} \int WD_{op}VN_{ip}^{-1}h\mathrm{d}\omega$$

$$= \inf_{Q\in A_0^\infty} \| WD_{op}VN_{ip}^{-1} + WD_{op}N_{op}Q \|_\infty = \max_{h\in BH^1} | < WD_{op}VN_{ip}^{-1}, \overline{h} > | \qquad (7.4)$$

and the existence of a maximizing $h \in H^1$, (h_o say), is assured. Since (7.4) is true there is no duality gap.

The above analysis will be exploited below.

7.2 Duality Theory in Approximation Theory

This section gives an approach for assessing how far the obtained solution is from the optimal solution.

Next the important concept of alignment is defined. Taking $x \in X$ where X is a vector space and $f \in X^*$, where X^* is the dual of X then x and f are said to be aligned if

$$< f, x > = \|x\|_X \; \|f\|_{X^*}$$

Any system with the following property of constant magnitude at all frequencies is termed all pass

$$|P(j\omega)| = k \;\; \forall \;\; \omega \;\text{ where } k \text{ is a constant}$$

It is known that alignment holds if a solution is optimal [23]. If X is taken to be

H^{∞}, the set of all stable transfer functions, then alignment effectively means *all pass.* Using this it could be observed from the Bode plot if the optimal solution has been obtained. If there is a gap between the optimal solution and the solution obtained from the Akilov and Rubinov algorithm there will not be alignment. If the optimal solution is not obtained a method for determining how far it is from the optimal and hence the cost of truncation would be valuable information.

It has been shown in work by R. C. Buck [8,9,10] that closeness to alignment is a necessary and sufficient condition for closeness to optimality. This was shown as follows.

Let X denote a (real) linear space with norm $\|\ \|$. Take X^* to be the dual space of continuous linear functionals. It is a Banach space with norm

$$\|f\| = \sup_{\|x\| \leq 1} |< f, x >| \quad \text{where} \quad f \in X^*$$

Again, let M denote a subspace of X. Let BX be the unit sphere in X and BX^* be the unit sphere in X^*. Let $BM^{\perp}$ be the unit ball in $M^{\perp}$.

Define three sets A, B and C as follows. The set A is almost aligned to within ϵ. The set B is almost optimal to within ϵ. The set C is almost aligned to within 2ϵ. Specifically,

$$A = \left\{x_0 - m; \|x_0 - m\| - \epsilon \leq < f, x_0 - m > \text{ some } f \in BM^{\perp}\right\}$$

$$B = \{x_0 - m; \|x_0 - m\| \leq \rho_M(x_0) + \epsilon\}$$

$$C = \left\{x_0 - m; \|x_0 - m\| - 2\epsilon \leq < f, x_0 - m > \text{ some } f \in BM^{\perp}\right\}$$

where $\rho_M(x_0)$ is defined as follows,

$$\rho_M(x_0) = \inf_{m \in M} \|x_0 - m\|$$

Theorem For the sets A, B and C as defined above, the following holds

$$x_0 - m \in A \Rightarrow x_0 - m \in B \Rightarrow x_0 - m \in C$$

□

This can be expressed informally in words as,

$$\text{Almost Aligned} \Leftrightarrow \text{Almost Optimal}$$

Proof :

Taking $x_0 - m \in A$,

$$x_0 - m \in A \Rightarrow \exists\ f \in BM^{\perp} \text{ such that } < f, x_0 - m > \geq \|x_0 - m\| - \epsilon$$

Also by definition,

$$\rho_M(x_0) = \inf_{m \in M} \|x_0 - m\|$$

From the Hahn-Banach theorem

$$\inf_{m \in M} \|x_0 - m\| = \max_{f \in M^{\perp}, \|f\| \leq 1} | < f, x_0 > |$$

So,

$$\rho_M(x_0) = \max_{f \in M^{\perp}, \|f\| \leq 1} | < f, x_0 > |$$

and

$$< f, x_0 - m > \leq \max | < f, x_0 > | = \rho_M(x_0)$$

Combining this with the definition of set A gives,

$$\Rightarrow \|x_0 - m\| \leq < f, x_0 - m > + \epsilon \leq \rho_M(x_0) + \epsilon$$

$$\Rightarrow x_0 - m \in B$$

so,

$$x_0 - m \in A \Rightarrow x_0 - m \in B \tag{7.5}$$

Now taking $x_0 - m \in B$,

$$x_0 - m \in B \Rightarrow \|x_0 - m\| \leq \rho_M(x_0) + \epsilon$$

Choose

$$f \in BM^{\perp} \text{ such that } < f, x_0 - m > \geq \rho_M(x_0) - \epsilon$$

Combining this with the definition of set B gives,

$$\Rightarrow \|x_0 - m\| \leq < f, x_0 - m > +2\epsilon$$

$$\Rightarrow x_0 - m \in C$$

so,

$$x_0 - m \in B \Rightarrow x_0 - m \in C \tag{7.6}$$

Combining both results (7.5) and (7.6) gives

$$x_0 - m \in A \Rightarrow x_0 - m \in B \Rightarrow x_0 - m \in C$$

as claimed. □

This establishes that closeness to alignment is a necessary and sufficient condition for closeness to optimality. This is a global result rather than a local result, meaning that ϵ does not have to be small. This result can be used in an analysis of the cost of truncation. By using this result it is possible to qualitatively assess the cost of H^∞ truncation. In order for this to be a quantitative approach it would be necessary to find a way to compute suitable f's. The next section will give a result which can be used to obtain a quantitative evaluation of the cost of H^∞ truncation.

7.3 Alignment with Akilov and Rubinov for WS example

The previous section has shown that closeness to alignment is a necessary and sufficient condition for closeness to optimality. While this is a very useful result it unfortunately doesn't give a lower bound on the true optimal solution. This section was motivated by the result from R. C. Buck that closeness to alignment gives some measure of optimality.

The Akilov and Rubinov algorithm uses a truncated subspace of H_∞. This means that the solution obtained is an upper bound for the true optimum as it is a finitely many variables problem. The following is a method of obtaining a lower bound on the true optimum. It is applied here to the $\|W_1S(Q)\|_\infty$ problem. A similar approach could be used for any of the other specifications in order to obtain a lower bound on the true optimum. The result obtained in this section means that bounds on the optimum can be obtained. This result can be used to effectively evaluate the issue of truncation of H^∞.

First, alignment in the context of this example is shown. If $W_1S(Q_1)$ where $Q_1 \in H^\infty$ and $h_1 \in H^1$ are aligned then the following relationship holds

$$\int W_1S(Q_1)h_1 d\omega = \|W_1S(Q_1)\|_\infty \|h_1\|_1$$

$$\Rightarrow \int |W_1S(Q_1)h_1| d\omega = \|W_1S(Q_1)\|_\infty \|h_1\|_1$$

$$\Rightarrow \int |W_1S(Q_1)h_1| d\omega = \|W_1S(Q_1)\|_\infty \int |h_1| d\omega \tag{7.7}$$

Since $Q_1 \in H^\infty$ and $h_1 \in H^1$ are aligned it follows that both are optimal. The general relationship which applies even if not aligned is

$$\int |W_1S(Q)h_1| d\omega \leq \|W_1S(Q)\|_\infty \int |h_1| d\omega$$

This above inequality is called Holder's inequality.

Equation (7.7) can only be true if $W_1S(Q)$ is all pass. Strictly speaking it is W_1S that is aligned with h_1, but W_1S is aligned for a particular Q. As stated previously, aligned effectively means all pass (constant magnitude for all frequencies).

Now, suppose that the true optimum λ is given by

$$\lambda = \sup_{h \in BH^1} \int W_1S(Q_1)h d\omega = \int W_1S(Q_1)h_1 d\omega$$

In the above expression $h_1 \in BH^1$ is the optimal $h \in BH^1$. Let Q_{AR} be the result for the Youla parameter obtained from the Akilov and Rubinov algorithm. This gives

$$\lambda \geq \int W_1 S(Q_{AR}) h_2 d\omega \ \ \forall\ h_2 \in BH^1$$

Suppose that Q_{AR} is optimal for some weight, say W_2. Then $W_2 S(Q_{AR})$ is aligned for some h, say h_3. Since they are aligned the following holds

$$\int W_2 S(Q_{AR}) h_3 d\omega = \int |W_2 S(Q_{AR}) h_3| d\omega = \|W_2 S(Q_{AR})\|_\infty \|h_3\|_1$$

By the all pass property it may be taken that

$$W_2 = \frac{1}{S_{op}(Q_{AR})}$$

Next choose

$$h_2 = \frac{\frac{W_2}{W_1} h_3}{\|\frac{W_2}{W_1} h_3\|_1}$$

giving

$$\lambda \geq \frac{\int |W_2 S(Q_{AR}) h_3| d\omega}{\|\frac{W_2}{W_1} h_3\|_1} \tag{7.8}$$

$$\Rightarrow \lambda \geq \frac{\|W_2 S(Q_{AR})\|_\infty \|h_3\|_1}{\|\frac{W_2}{W_1} h_3\|_1}$$

Using the fact that $\|\frac{W_2}{W_1} h_3\|_1 \leq \|\frac{W_2}{W_1}\|_\infty \|h_3\|_1$, gives

$$\lambda \geq \frac{\|W_2 S(Q_{AR})\|_\infty}{\|\frac{W_2}{W_1}\|_\infty}$$

$$\Rightarrow \lambda \left\|\frac{W_2}{W_1}\right\|_\infty \geq \|W_2 S(Q_{AR})\|_\infty$$

$$\Rightarrow \lambda \left\|\frac{W_2}{W_1}\right\|_\infty \geq 1$$

$$\Rightarrow \lambda \left\|\frac{1}{W_1 S(Q_{AR})}\right\|_\infty \geq 1$$

$$\Rightarrow \lambda \geq \min_\omega |W_1 S(Q_{AR})|$$

This gives a lower bound for λ. So this means that the following inequality holds

$$\lambda_{akrub} \geq \lambda_{true} \geq \min_{\omega} |W_1 S(Q_{AR})|$$

This means that the optimal solution is contained between the minimum and maximum of $|W_1 S(Q_{AR})|$. This is an intuitively pleasing result because it means that the closer the minimum and maximum values the smaller the range by which the optimum differs from the Akilov and Rubinov obtained solution. This is intuitive as it effectively means that the closer the solution is to all pass the closer the obtained solution is to the optimum.

7.4 Basis Selection

This section states techniques used for selection of basis functions and discusses the issue.

There are two main issues involved in selecting the basis for the subspace of H^∞.

Pole selection One of the observations to make would be that a peak at a particular frequency means that a basis function with a pole at this frequency may result in a solution which is closer to the true optimum. The approach of Sections 7.2 and 7.3 may provide a useful addition to the basis selection procedure.

Number of terms By observing how close to the optimum the obtained solution lies will give a strong indication of the number of terms required in the basis functions.

Boyd *et al.* [5] suggested that a good choice for the structure of $Q(s)$ is a linear combination of simple stable transfer functions $q_i(s)$ of the form

$$Q(s) = \sum_{i=1}^{n} a_i q_i$$

Boyd *et al.* [7] on the two-disk problem employed a second method for selecting their basis functions. Using the bilinear transformation

$$s = 20\frac{z+1}{z-1}$$

they mapped the solution space to discrete time. He then assumed that the optimal parameter $Q(z)$ is closely approximated by a finite impulse response (FIR) filter with 20 taps. It was found that increasing the number of taps did not significantly improve the solution having found a 20-tap Q which satisfies the constraints.

In work done by Webers and Engell [27] they outline that the choice of base functions for the series expansion of the Youla parameter is crucial for its success.

In the present author's numerical work, the basis functions were chosen as follows

$$Q(s) = \sum_{i=1}^{n} \alpha_i y_i$$

where

$$y_i = \frac{s^{i-1}}{(s+\omega_c)^n} \qquad i = 1, ..., n$$

This can be interpreted as a Taylor series expansion about $\omega = \omega_c$. Here ω_c is the target crossover frequency, which is the crucial region for a design.

7.5 Examples

In this section the bounds described above on the optimal solution are obtained for two problems. The first problem has a known analytical solution and the second is an analytically unsolved problem. The first problem was given in Chapter 5 as part of the software validation, but is used here to confirm the procedure adopted in the quantitative analysis of truncation.

Specifically, consider finding the controller which minimizes

$$\inf \| \ |W_1 S| \ \|_\infty$$

where, as usual,

$$S = \frac{1}{1+L}, \qquad\qquad T = \frac{L}{1+L}$$

L is the loop gain, and W_1 is a weighting function. The minimization is over all feedback controllers which stabilize the closed loop system.

The plant is

$$G = \frac{-s+3}{-s+4}$$

and the weight is

$$W_1 = \frac{s+12}{s+1}$$

Applying the algorithm to this problem yields the results given below in Table 7.1 and Figures 7.1 to 7.5.

n	max	min
2	43.7	9.7
3	31.7	14.8
4	28.7	17.9
5	27.5	20.7
6	26.9	23.0
7	26.6	24.4
8	26.4	24.9
9	26.37	25.5
10	26.33	26.03
11	26.32	26.06
12	26.29	26.2

Table 7.1: $\max_{\omega} |W_1 S|$ and $\min_{\omega} |W_1 S|$ for various n's.

To illustrate the utility of the results the method is applied to a problem with no known analytical solution.

Specifically, consider finding the controller which minimizes

$$\inf \| \ |W_1 S| + |W_2 T| \ \|_\infty$$

The plant is

$$G = \frac{-s+3}{-s+4}$$

and the weights are

$$W_1 = \frac{s+12}{s+1} \qquad\qquad W_2 = 1$$

Applying the algorithm to this problem yields the results given below in Table 7.2 and Figures 7.6 to 7.9.

n	max	min
5	35.0	32.8
10	34.6	34.52
15	34.59	34.55

Table 7.2: $\max_\omega \ |W_1 S| + |W_2 T|$ and $\min_\omega \ |W_1 S| + |W_2 T|$ for various n's.

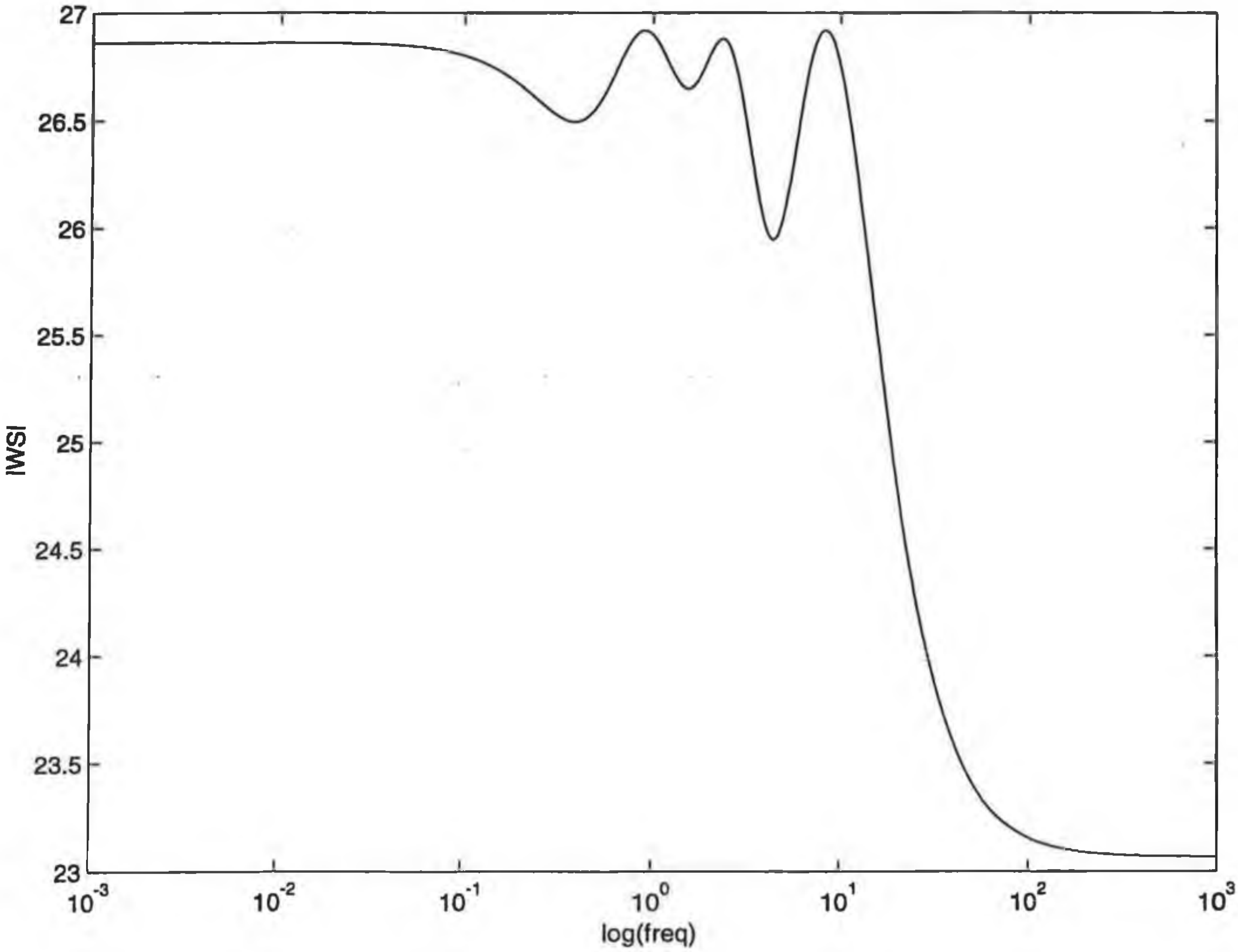

Figure 7.1: $|WS|$ for $n = 6$, truncation issue, example 1

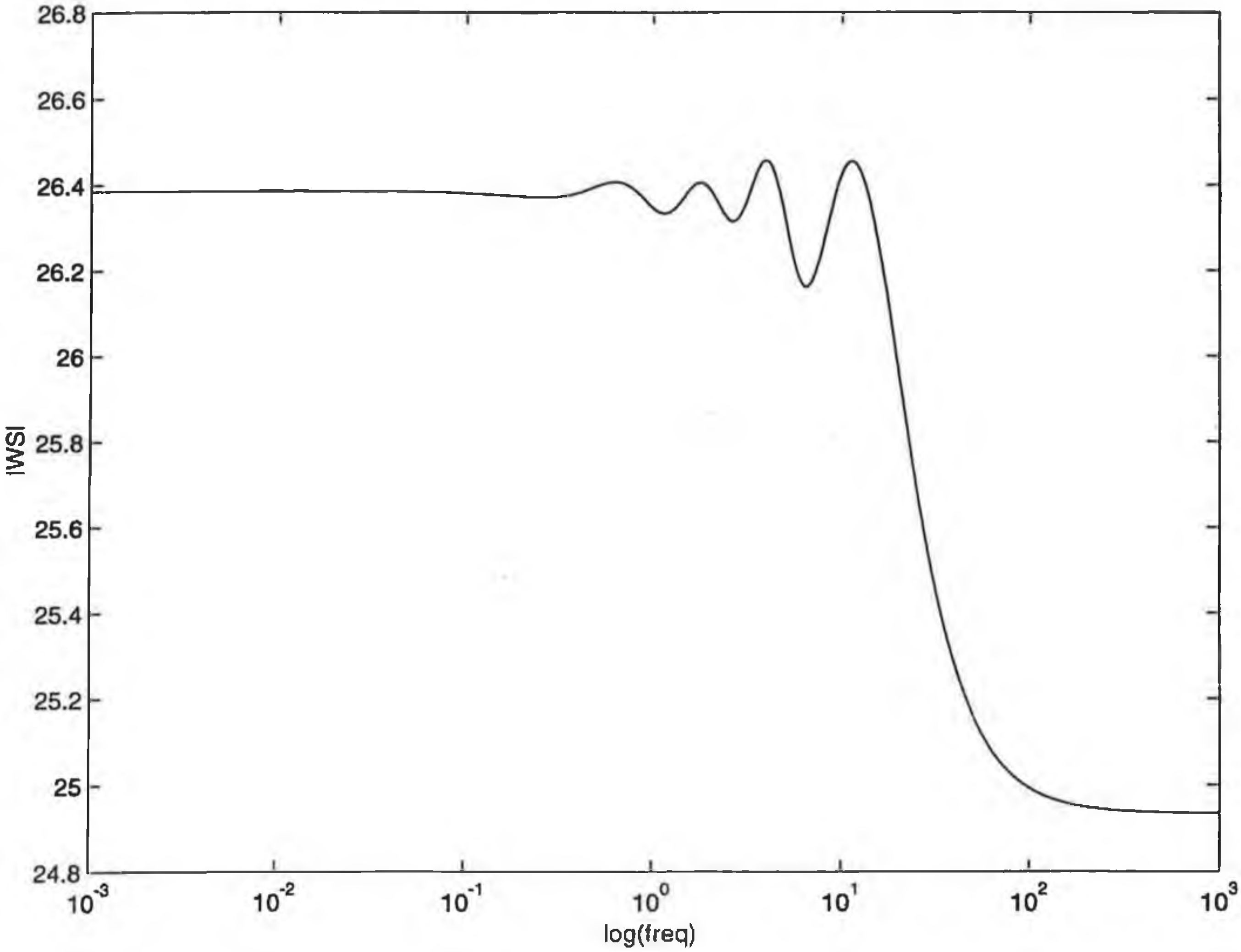

Figure 7.2: $|WS|$ for $n = 8$, truncation issue, example 1

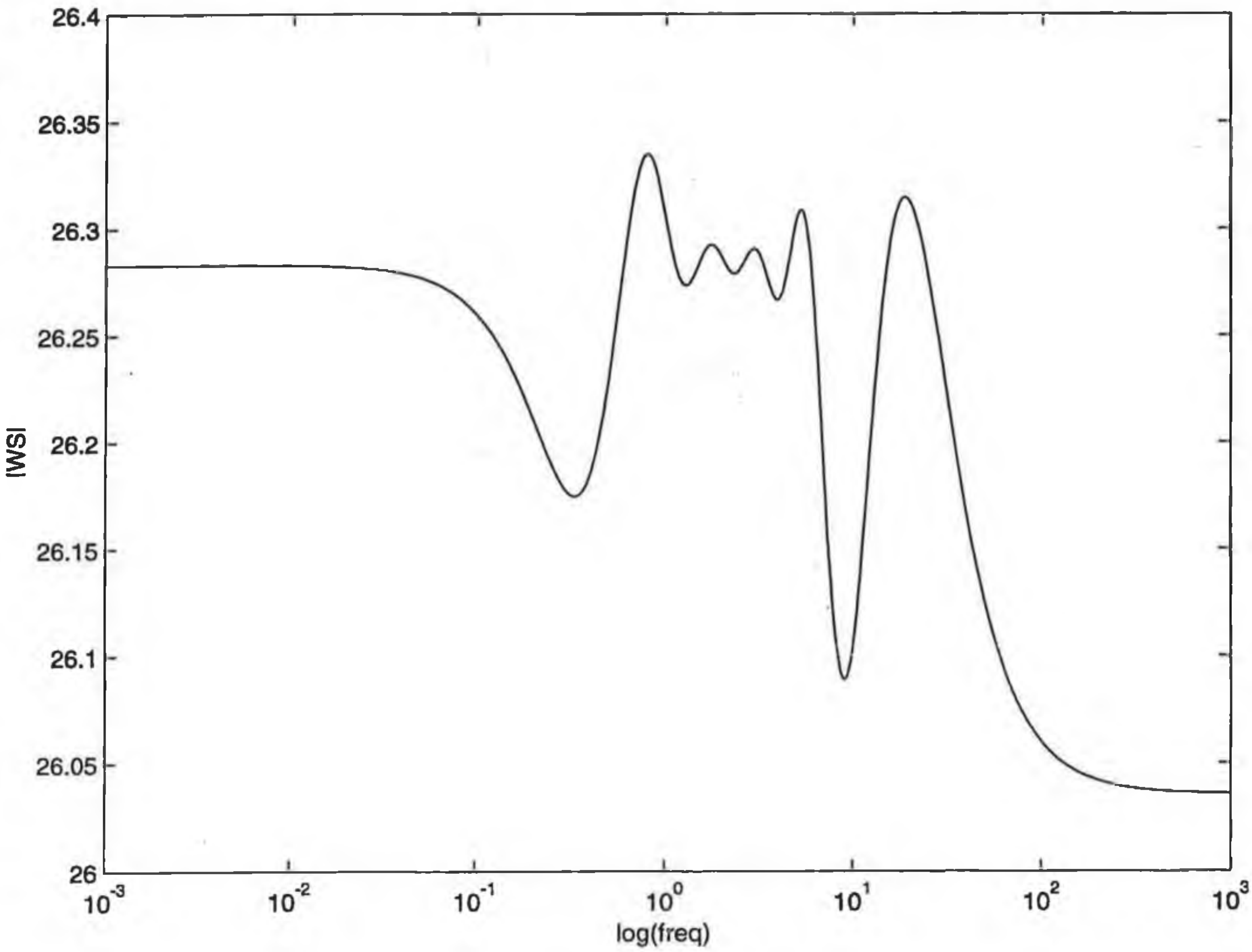

Figure 7.3: $|WS|$ for $n = 10$, truncation issue, example 1

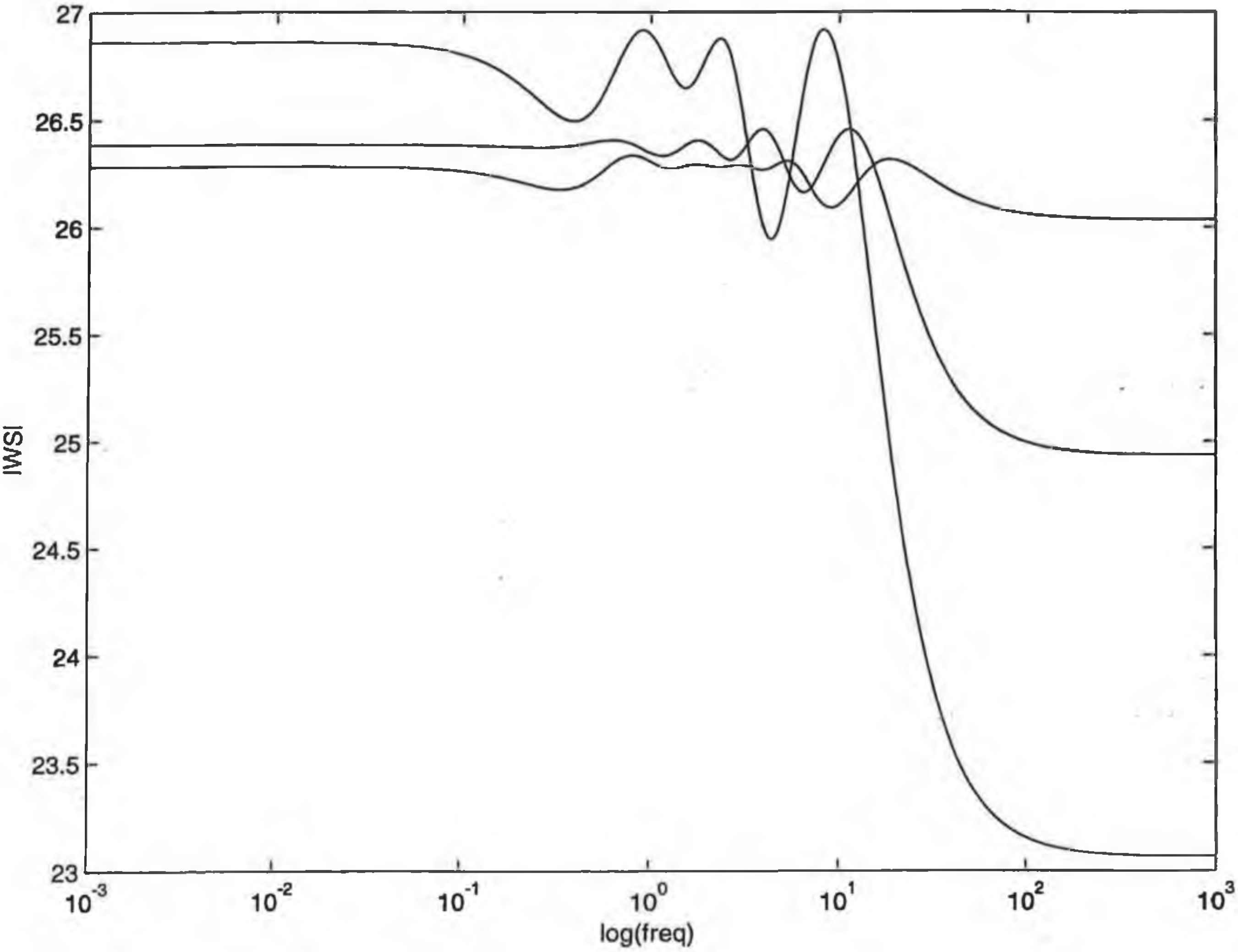

Figure 7.4: $|WS|$, for various n's,truncation issue, example 1

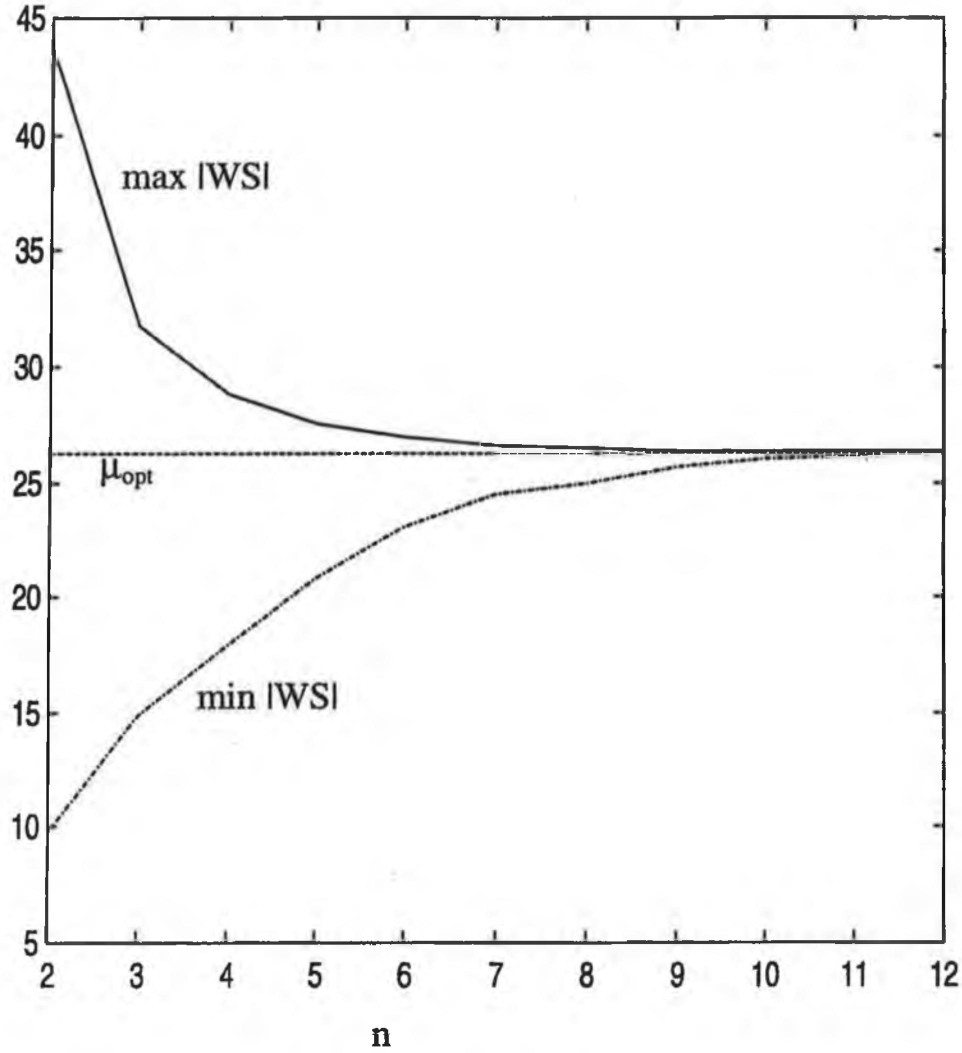

Figure 7.5: $\max_{\omega} |WS|$, μ_{opt} and $\min_{\omega} |WS|$ for various n's, truncation issue, example 1

From the data it can be seen that as the number of terms in the basis increases the problem gives a solution which is closer to all-pass. Using R. C. Buck's result this shows that the quality of the obtained solution is approaching optimality. This gives a qualitative evaluation of the result.

In addition to this a quantitative measure of how close the obtained solution λ_{akrub} (from the truncated vector space) is to the true optimum λ_{true} (from the infinite dimensional vector space). For each of the examples with differing numbers of terms in the basis an upper and a lower bound on the true optimum is given. In general, one would expect that tighter bounds are obtained as the number of terms increases. In both examples this is in fact the case.

For the problem where the true optimum is given by

$$\lambda_{true} = \inf_{Q \in H^{\infty}} \|W_1 S(Q)\|_{\infty}$$

the following bounds apply

$$\lambda_{akrub} \geq \lambda_{true} \geq \min_{\omega} |W_1 S(Q_{AR})|$$

The bounds approached the known analytical solution as the number of basis elements increased. This is as expected.

For the problem where the true optimum is given by

$$\lambda_{true} = \inf_{Q \in H^{\infty}} \| \, |W_1 S(Q)| + |W_2 T(Q)| \, \|_{\infty}$$

the following bounds apply

$$\lambda_{akrub} \geq \lambda_{true} \geq \min_{\omega} |W_1 S(Q_{AR})| + |W_2 T(Q_{AR})|$$

It can clearly be seen that the bounds approached a specific value as the number of basis elements increased. Again, this is as expected. This suggests that the true optimal solution for this problem is $\lambda_{true} = 34.57 \pm 0.02$.

These examples illustrate the usefulness of these bounds.

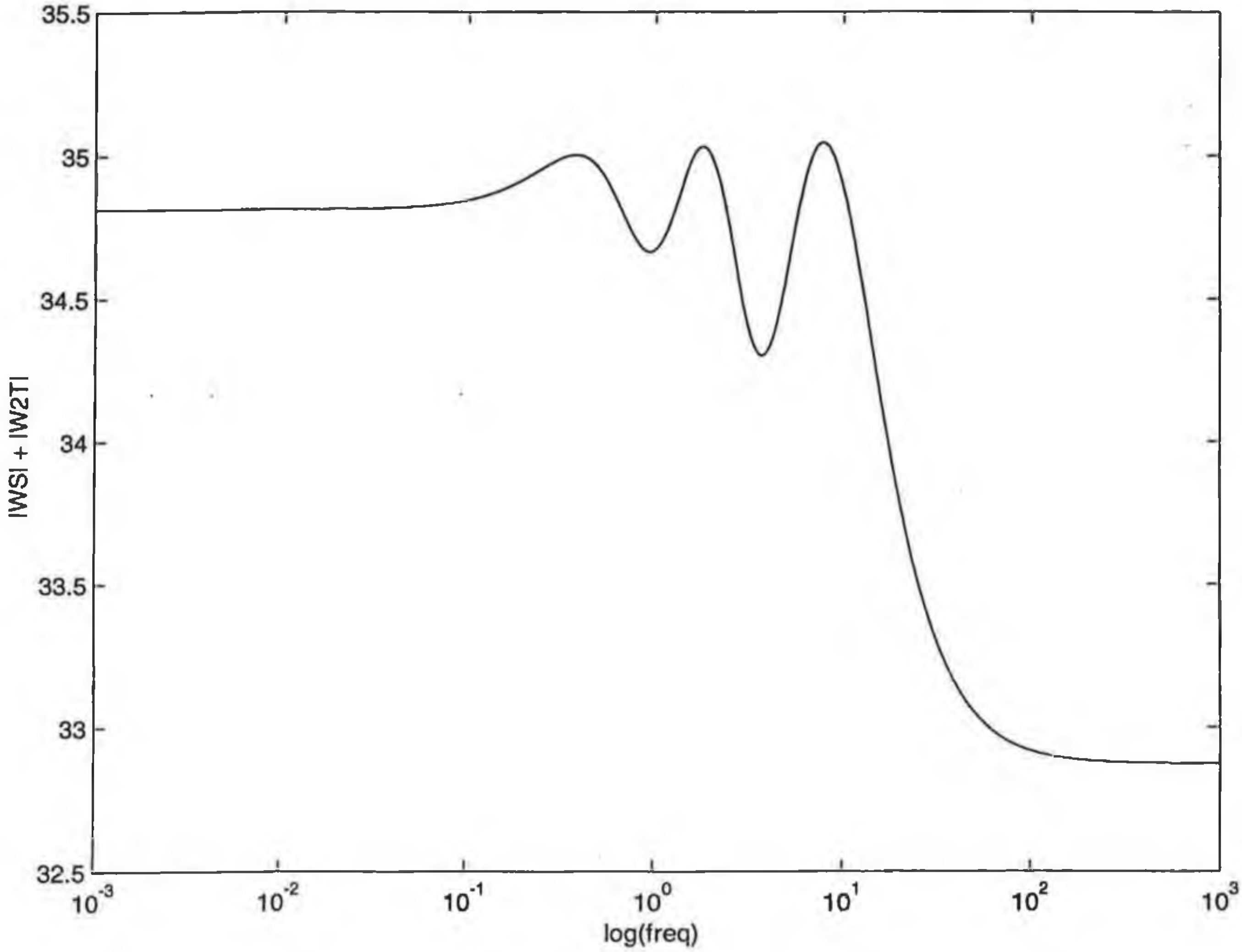

Figure 7.6: $|W_1 S| + |W_2 T|$ for $n = 5$, truncation issue, example 2

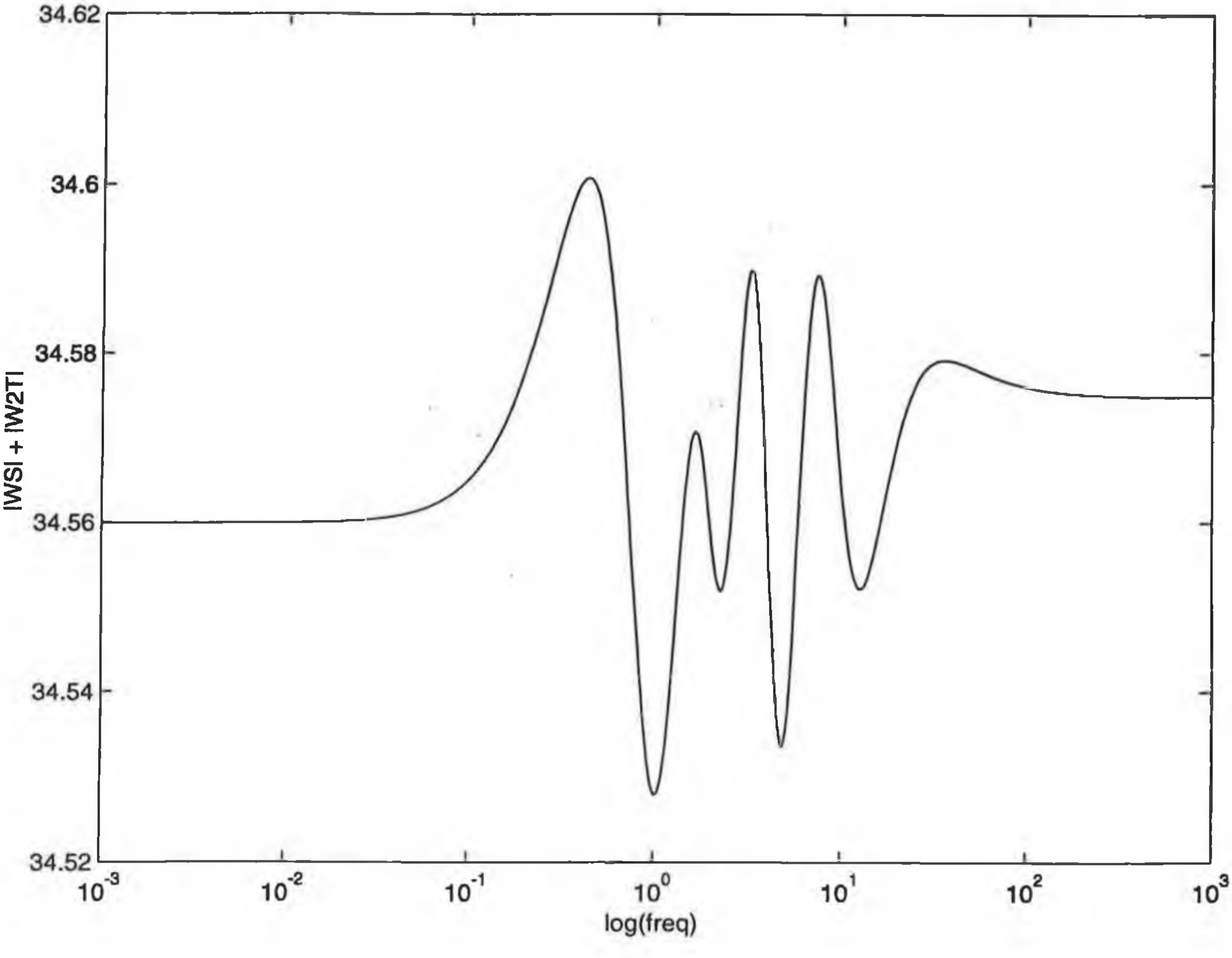

Figure 7.7: $|W_1S| + |W_2T|$ for $n = 10$, truncation issue, example 2

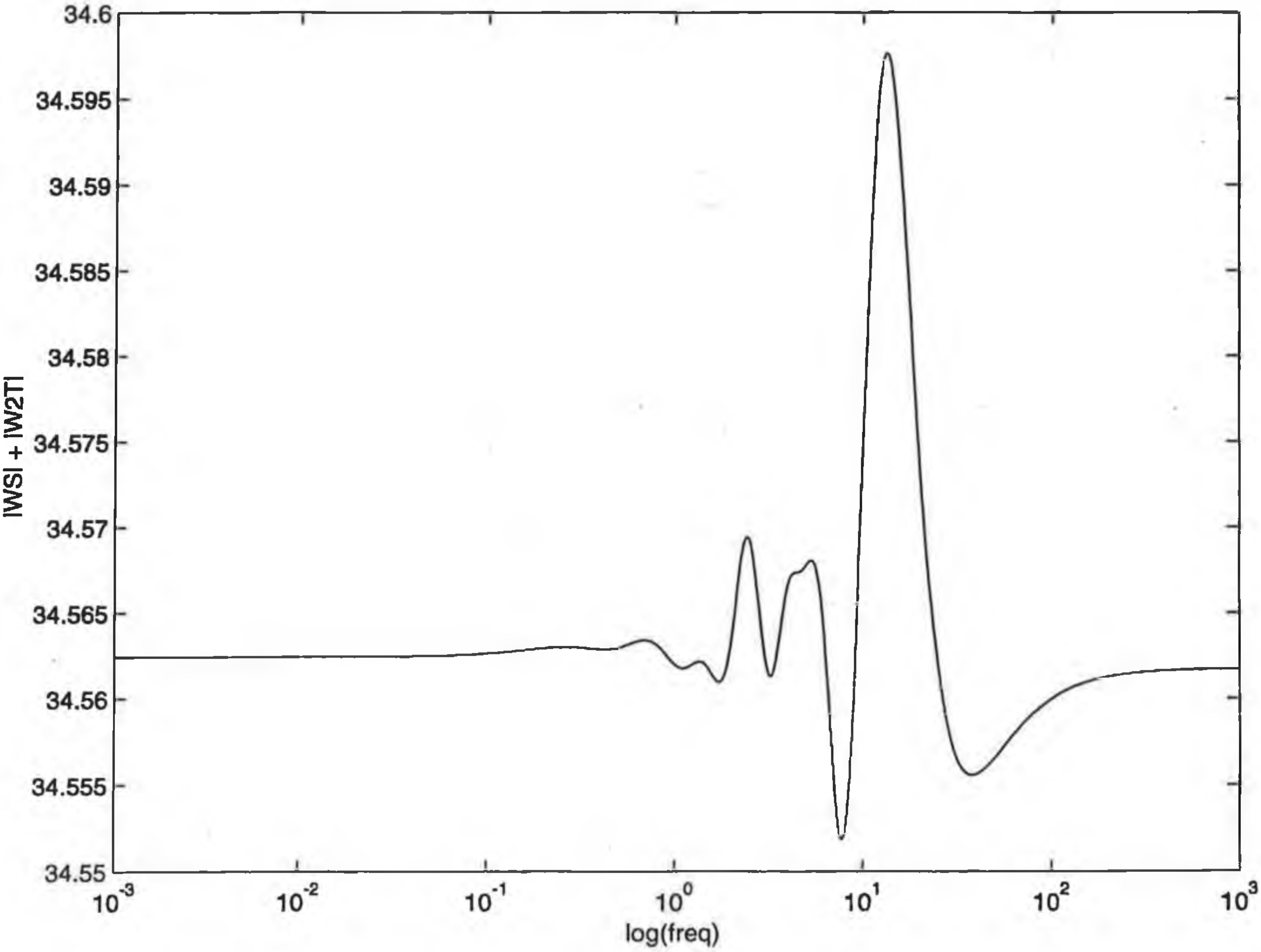

Figure 7.8: $|W_1S| + |W_2T|$ for $n = 15$, truncation issue, example 2

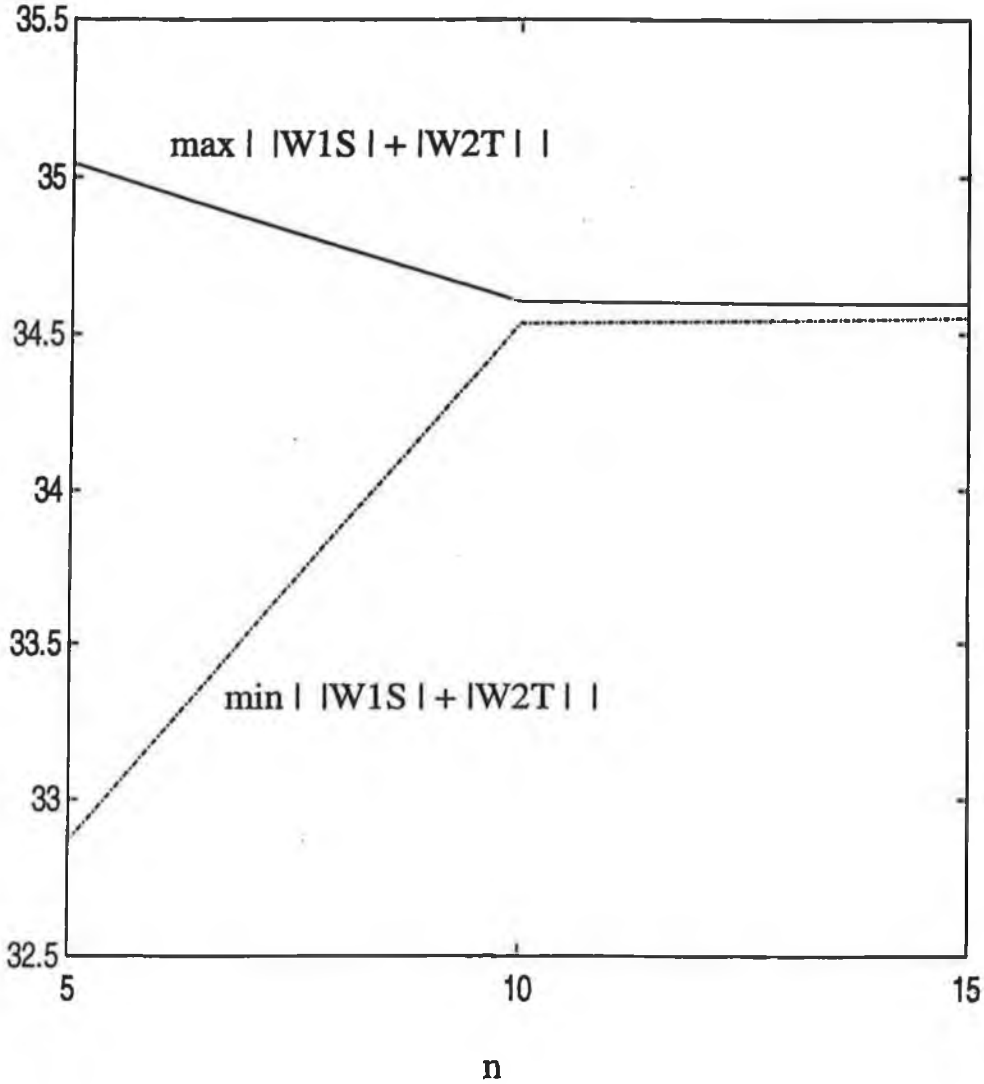

Figure 7.9: $\max_{\omega} \; |W_1S| + |W_2T|$ and $\min_{\omega} \; |W_1S| + |W_2T|$ for various n's, truncation issue, example 2

7.6 Remarks

Using the result outlined in the previous sections the gap between the optimal solution of the infinite problem, λ_t^p, and the solution obtained using the Akilov and Rubinov algorithm can be measured. By plotting $|x_0 - m|$ against ω and observing how close it is to alignment (all pass) a measure of the difference between it and the optimum can be obtained.

The duality theory approach to obtain a lower bound and a measure of closeness to optimality is applied to the $||WS||_\infty$ problem. This example was used to demonstrate a possible evaluation of the cost of truncation. The analysis proposed here applies equally well to the robust disturbance attenuation problem and the two disc problem because the all pass property applies [19,20]. Note that these problems have not been solved analytically. This provides a qualitative and quantitative analysis of the issue of truncating H^∞.

Chapter 8

Conclusions

In this thesis, improvements were made on the Boyd-Barratt paradigm for feedback controller design. An alternative numerical approach with a number of advantages was adopted. Also, a qualitative and quantitative evaluation of the largest disadvantage of the Boyd-Barratt paradigm, namely the required truncation of H^∞, was given.

The thesis began by outlining the Boyd-Barratt paradigm for feedback controller design. It was outlined how this approach combines the Youla parameterization with convex optimization. A completely different numerical approach was adopted, but otherwise their paradigm is accepted in its entirety.

It was shown how control problems can be recast as linear programs by use of Youla parameterisation. It was shown how problems which are non-convex in the controller become convex problems via use of the Youla parameter. However there is no guarantee on the accuracy of results obtained from these finite linear programs. There seems to be no reason to be optimistic that the controller produced by this approach will be close to the true optimal controller. Also, there is the difficulty of deciding how large a linear program to select, and how to choose the subset of constraints to be retained. This suggested seeking a more sophisticated approach.

The new numerical scheme adopted involved using an algorithm due to Akilov and Rubinov. This completely circumvented the need to compute derivatives or subdifferentials, which can be a difficult task. Instead, certain linear functionals were computed, and this is generally quite straightforward. The algorithm made use of a linear equation solver and a linear program solver, which are standard numerical problems. This resulted in code which is easier to implement, much shorter and more elegant than that required to compute complicated gradients. The coding of the algorithm gave very promising computing times and it is felt that this justifies the approach. The approach also has the attractive feature of giving bounds at each iteration, which assist in convergence monitoring.

The Boyd-Barratt paradigm has the disadvantage that an infinite dimensional Banach space must be truncated to a finite dimensional subspace prior to optimizing. This thesis also applies certain primal-dual techniques from functional analysis to study the implications of this truncation. Primal-dual theory is used to show that the true optimal solution lies within the solution of two semi-infinite linear programming problems, namely the dual problem with finitely many variables and the primal problem with finitely many variables.

Also, results due to R. C. Buck have been used to show that nearness to alignment gives a qualitative indication of nearness to optimality. A quantitative indication has also been developed. The analysis proposed here applies equally well to the robust disturbance attenuation problem and the two disc problem. The results were illustrated with examples.

In conclusion, the use of the Akilov and Rubinov algorithm is an improvement on the previous numerical approach used by Boyd-Barratt. In addition, the thesis gave an effective analysis of the truncation to a finite dimensional subspace prior to optimizing, which is the largest shortcoming of the Boyd-Barratt approach. The thesis demonstrates how to obtain a qualitative and quantitative indication of the cost of this truncation.

References

1. L. Ahlfors, *Complex Analysis*, 3^{rd}-Edition, McGraw-Hill, 1979.

2. G. P. Akilov and A. M. Rubinov, "The Method of Successive Approximations for Determining the Polynomial of Best Approximation", *Soviet Math. Dokl.*, Vol. 5, pp. 951-953, 1964.

3. M. J. Best and K. Ritter, *Linear Programming*, Prentice-Hall, pp. 118-137, 1985.

4. S. P. Boyd, C. H. Barratt and S. Norman, "A New CAD Method and Associated Architectures for Linear Controllers", *Proc. IEEE*, Vol. 33, No. 3, pp. 268-282, Mar. 1988.

5. S. P. Boyd and C. H. Barratt, *Linear Controller Design - Limits of Performance*, Prentice Hall, New Jersey, 1991.

6. S. P. Boyd, C. H. Barratt and S. Norman, "Linear Controller Design: Limits of Performance via Convex Optimization", *Proc. IEEE*, Vol. 78, No. 3, pp. 529-574, Mar. 1990.

7. S. P. Boyd, S. Norman, "Numerical Solution of a Two-Disk Problem",

8. R. C. Buck, "Applications of Duality in Approximation Theory", *Proc. Symp on Approx. of Functions*, General Motors Research Labs, pp. 27-42, 1964.

9. R. C. Buck, "Linear Spaces and Approximation Theory", University of Wisconsin, Madison.

10. R. C. Buck, "A Complete Characterization for Extreme Functionals", University of Wisconsin, Madison.

11. E. W. Cheney, *Introduction to approximation theory*, McGraw Hill, New York, pp. 72-100, 1966.

12. A. D. Dahleh , I.J. Diaz-Bobillo, *Control of Uncertain Systems*, Prentice Hall.

13. R. W. Daniels, *Approximation Methods for Electronic Filter Design*, McGraw Hill, New York, 1974.

14. C. A. Desoer, R.-W. Liu, J. Murray and R. Saeks, "Feedback System Design: The Fractional Representation Approach to Analysis and Synthesis", *IEEE Transactions on Automatic Control*, Vol. 25, No. 6, May 1980.

15. J. A. Donnellan, A. M. Holohan, " Convex Optimization for Controller Design", *Proc. ISSC*, June. 2000.

16. P. Duren, *Theory of* $\mathcal{H}_p$ *Spaces*, Academic Press, New York, 1970.

17. J. B. Garnett, *Bounded Analytic Functions*, Academic Press, 1981.

18. A. M. Holohan, G. Meinsma, "Inverse Problems in 2-Norm and Infinity-Norm Controller Synthesis", Memo No. 1459, University of Twente, Aug. 1998.

19. A. M. Holohan, M. G. Safonov, "Neoclassical Control Theory : A Functional Analysis Approach to Optimal Frequency Domain Controller Synthesis", in *Control and Dynamic Systems*, Vol 50, Academic Press, 1992.

20. A. M. Holohan, M. G. Safonov, "Duality Relations for the Optimal Two-Disk H^∞ problem", *Proc. ACC*, Chicago, pp. 1844-1849, June. 1992.

21. P. Koosis, *Introduction to $\mathcal{H}_p$ Spaces*, London Mathematical Society Lecture Note Series, Vol. 40, Cambridge University Press, 1980.

22. D. G. Luenberger, *Linear and Non-Linear Programming*, Addison-Wesley, Massachusetts, 1984.

23. D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, 1969.

24. E. Y. Remes, "Theorie Des Fonctions", *Seances de L'Academie des Sciences*, Paris, Vol. 198, pp. 2063-2065, 1934.

25. E. Y. Remes, "Sur le Calcul Effectif Des Polynomes D'approximation de Tchebiches", *Seances de L'Academie des Sciences*, Paris, Vol. 199, pp. 337-340, 1934.

26. W. W. Rogosinski and H. S. Shapiro, "On Certain Extremum Problems for Analytic Functions", *Acta Math.*, Vol. 90, pp. 287-318, 1953.

27. C. Webers, S. Engell "Design of Optimal Controllers with Respect to Time and Frequency Domain Specifications", *Proc. IFAC*, Smolenice, pp. 215-219, Sept. 1994.

28. D. C. Youla, J. J. Bongiorno and H. A. Jabr, "Modern Wiener-Hopf Design of Optimal Controllers. Part I", *Proc. IEEE*, Vol. AC-21, pp. 3-13, June 1976.

29. D. C. Youla, H. A. Jabr and J. J. Bongiorno, "Modern Wiener-Hopf Design of Optimal Controllers. Part II", *Proc. IEEE*, Vol. AC-21, pp. 319-338, June 1976.

30. G. Zames and B. A. Francis, "Feedback, Minimax Sensitivity, and Optimal Robustness", *IEEE Transactions on Automatic Control*, Vol. 28, No. 5, pp. 585-601, May 1983.