An Algorithm to Compute the Distance to Uncontrollability of Linear Time-Invariant Control Systems

by

© *Kennedy Azupwah*

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of *Science*

Scientific Computing Program

Memorial University of Newfoundland

May 2017

St. John's Newfoundland

Abstract

In this study, we determine how far a Linear Dynamic System is from the nearest uncontrollable system. We will call this quantity "The Distance to Uncontrollability". Estimating this distance, not only do we know if a given linear dynamical system is controllable or uncontrollable, but in the case of a controllable system, we also know how far it is from being uncontrollable. This could be found useful by a control engineer for example, in making a decision to insert additional controls to the system design.

As it turns out, the estimation of the "distance to uncontrollability" is equivalent to determining the global minimum of a certain function. In this work, we will examine some already existing algorithms and will present a Two-Phase Algorithm that will combine a novel search algorithm, termed the Density Search Algorithm and the Tunneling Algorithm [21] for the computation of this global minimum.

— MUN School of Graduate Studies

Acknowledgements

Foremost, I would like to give thanks to God almighty for the gift of Life. I would also like to express my sincere gratitude to my thesis advisor Prof. Miminis of the Faculty of Computer Science at Memorial University for his continuing support. You are the most patient man I know. The door to his office was always open whenever I ran into trouble or had a question about my research. Your immense knowledge and guidance helped me greatly during this research. He consistently allowed this paper to be my own work but steered me in the right direction whenever he thought I needed it. This research accomplishment would not have been possible without him.

Last, not the least, I would like to thank my Mother for her continues to support and her prayers. Thank you

— MUN School of Graduate Studies

Contents

List of Tables

List of Figures

1 Introduction

1.1 Basic Definitions

Definition 1 (Vector Norm) *Given $x \in \mathbb{C}^n$, the Euclidean norm of x is*

$$\|x\|_2 = \left(x^H x\right)^{\frac{1}{2}}$$

where x^H is the complex conjugate transpose of x.

Definition 2 (Matrix Norm) *Given $A \in \mathbb{C}^{m \times n}$, then*

$$\|A\|_2 = \max_{\|x\|_2 \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\|y\|_2 = 1} \|Ay\|_2$$

where $x, y \in \mathbb{C}^n$.

Definition 3 (Spectrum of a Matrix) *The set of eigenvalues of the matrix A, denoted by $\lambda(A)$, is known as its spectrum.*

Definition 4 (Span) *The span of vectors $u_i \in \mathbb{C}^n$, with $i = 1 : m$ is defined as the set of all linear combination of these vectors. That is*

$$\operatorname{span}\{u_1, u_2, \ldots, u_m\} = \{x : x = \eta_1 u_1 + \eta_2 u_2 + \cdots + \eta_m u_m \text{ for some } \eta_i \in \mathbb{C}\}$$

Definition 5 (Linear Independence) *The vectors $u_i \in \mathbb{C}^n$, $i = 1 : m$ are linearly independent, if and only if for $\forall \eta_i \in \mathbb{C}^n$, $i = 1 : m$ we have,*

$$\sum_{i=1}^{m} \eta_i u_i = 0 \;\Rightarrow \eta_1 = \eta_2 = \cdots = \eta_m = 0$$

Otherwise, the vectors are linearly dependent.

Definition 6 (Null Space) *Given a matrix $A \in \mathbb{C}^{m \times n}$, the null space of A is*

$$\mathcal{N}(A) = \{x : x \in \mathbb{C}^n \wedge Ax = 0\}$$

The dimension of $\mathcal{N}(A)$ is called the nullity of A and is denoted by $\operatorname{null}(A)$ *($\operatorname{null}(A) = \dim(A)$).*

Definition 7 (Rank) *Given a matrix $A \in \mathbb{C}^{m\times n}$, the column space of A, written $\mathcal{R}(A)$, is a subspace of $\mathbb{C}^n$ spanned by the columns of A. The row space of A is $\mathcal{R}\left(A^T\right) \subseteq \mathbb{C}^m$. The rank of A, written $\operatorname{rank}(A)$, is the dimension of $\mathcal{R}(A)$ or $\mathcal{R}\left(A^T\right)$, that is $\operatorname{rank}(A) = \dim \mathcal{R}(A) = \dim \mathcal{R}\left(A^T\right)$.*

Definition 8 (Orthogonality) *Two vectors $u, v \in \mathbb{C}^n$, are orthogonal if and only if $u^H v = 0$. If in addition $\|u\|_2 = \|v\|_2 = 1$ then, u and v are called orthonormal.*

Definition 9 (Orthogonal Matrices) *A matrix $U \in \mathbb{C}^{m\times n}$, with $m \geq n$ is called orthonormal if $U^H U = I_n$, where I_n is the identity matrix of size n. If $m = n$ then U is called unitary. If U is real, it will be called orthogonal.*

Theorem 10 *[30][p.318] Given a matrix $A \in \mathbb{C}^{m\times n}$, with $m \geq n$ and $\operatorname{rank}(A) = r \leq n$, we can compute unitary matrices U, V such that*

$$U^H A V = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}$$

with $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ and σ_i, $i = 1 : r$ are the singular values of A, while the columns of U and V are the left and right singular vectors of A respectively.

1.2 Preliminaries

We live in a four dimensional world of the space-time continua. We have all developed an awareness of the fact, that as we travel through space and/or time, things change. When something changes with respect to time then we say it is *dynamic*.

Science often attempts to represent complex phenomena which results in the derivation of a mathematical model that consists of interrelating equations. For example, a model of population growth would likely be dependent on such variables as food supply, population density, migration, etc. This set of equations is referred to as a *system*.

1.2.1 Continuous Time Systems

If one needs to represent a natural phenomenon that changes continuously with respect to time t, one may use a differential equation. An example of a continuous time dynamic phenomenon is the heating of a building.

A mathematical model that may describe the above phenomenon is

$$\dot{x}(t) = Ax(t)$$

where $A \in \mathbb{R}^{n\times n}$, depends on the characteristics of the phenomenon, $x(t) \in \mathbb{R}^n$ is the state of the phenomenon at the time t, and $\dot{x}(t)$ the derivative of $x(t)$.

1.2.2 Discrete Time Systems

Some natural phenomena may need not be described by a continuous time system. Consider a simple digital circuit. The state of the system only changes when we reach the end of a step, usually indicated by a clock pulse. So for this phenomenon we may need to use a difference equation of the form

$$x_{k+1} = Ax_k, \tag{1.1}$$

where x_k, x_{k+1} are the states of the above phenomenon at steps k and $k+1$ respectively. System (1.1) is known as discrete time system. Since the system is either in step k or $k+1$, it is not constantly changing like the continuous case.

1.2.3 Controllability of Dynamic Systems

Consider the continuous and discrete time systems respectively

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.2}$$

$$x_{k+1} = Ax_k + Bu_k \tag{1.3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$. The matrix A is known as the *system matrix* and B as the *input matrix* (other outside influences - for example readings from outside device or switches). In this system we say there are n states and m inputs. Also $x \in \mathbb{R}^n$, and $u \in \mathbb{R}^m$ are the state vector, and the control input of the system respectively at time t or at instance k. Often the two systems above will be referred to as (B, A), in which case any results will hold for both systems.
A fundamental concept of systems (1.2) and (1.3) is the concept of controllability which intuitively can be given by the following definitions.

Definition 11 (Controllability) *A continuous time system (1.2) is said to be* ***controllable*** *if for any pair of vectors $(x_i, x_f) \in \mathbb{R}^n \times \mathbb{R}^n$ there exists finite time $t > 0$, control $u(t)$ defined on the interval $[0, t]$, such that (1.2) with initial condition $x(0) = x_i$, produces solution $x(t) = x_f$.*

Definition 12 (Controllability) *A discrete time system (1.3) is said to be* ***controllable*** *if for any pair of vectors $(x_i, x_f) \in \mathbb{R}^n \times \mathbb{R}^n$ there exists a finite number of steps $k > 0$, controls $u_1, u_2, \ldots, u_k$, such that (1.3) with initial condition $x_0 = x_i$, produces solution $x_k = x_f$.*

Paige in [27] describe several tests to verify the controllability of systems (1.2) and (1.3) in the theorem below.

Theorem 13 *It can be shown, see [27] for example, that the system (B, A) is controllable if and only if one of the three conditions holds:*

1. $\operatorname{rank}(B, AB, A^2B, \ldots, A^{n-1}B) = n$
2. $\operatorname{rank}(B, A - \lambda I) = n$, $\forall \lambda \in \lambda(A)$
3. $\exists F \in \mathbb{R}^{m\times n} : \lambda(A) \cap \lambda(A + BF) = \emptyset$

From Theorem 13 above, it can be seen that the controllability definition is either a YES or NO answer; i.e, a system is either controllable or not. In practical applications where there are many uncertainties in the system model such as those resulting from modeling, linearization,

discretization, and other numerical and/or approximation errors, a controllable system may in fact be "almost" uncontrollable when these uncertainties are accounted for. So a measure of the "distance" to the nearest uncontrollable system would be more informative and desirable than the traditional "yes/no".

1.2.4 Computational Problems of Controllability

Each of the above methods in Theorem (13) has its own challenges when executed using finite precision arithmetic. Numerical errors can play a role in determining the rank of a matrix (as in the first two conditions) and in determining the equality of two eigenvalues (as in the last condition). For example, the approach in conditions (2) and (3) depends on finding eigenvalues, and so it is important to consider the sensitivity of eigenvalues in general, see Paige [27].

1.3 The Distance From an Uncontrollable System

Having the distance of a controllable system from the nearest uncontrollable is of far greater use than knowing that a system is simply controllable. Earlier theories such as theorem (13) above only provided us with a "yes" or "no" answer to whether the system is controllable or not. However, we would like to be able to determine if the system is "poorly controllable ". Hence the following definition, is more suitable for "measuring" controllability.

Definition 14 *Let $\mathcal{U}$ represent the set of uncontrollable systems and consider the system (B, A). We define*

$$\mu = \min_{(\delta B, \delta A) \in \mathbb{C}^{n \times (m+n)}} \{ \|(\delta B, \delta A)\|_2 : (B + \delta B, A + \delta A) \in \mathcal{U} \}$$

to be the distance of (B, A) from the nearest uncontrollable system. Clearly when $\mu = 0$ the system is uncontrollable.

We present a theorem that will give us the minimum needed perturbation to a matrix in order to decrease its rank making reference to the work in [24] . But before we present the theorem, we

first need the following two lemmas.

Lemma 15 *[26][p.237] Assume $A \in \mathbb{C}^{m\times n}$ with $r = \operatorname{rank}(A)$. Then the null space of A is a subspace of $\mathbb{C}^n$ of dimension $n - r$.*

Corollary 16 *Assume $A \in \mathbb{C}^{m\times n}$ with $r = \operatorname{rank}(A)$, then*

$$\operatorname{null}(A) + \operatorname{rank}(A) = n$$

Lemma 17 *If $y \in \operatorname{span}\{u_1, u_2, \ldots, u_n\}$, where the vectors u_i, $i \in \{1 : n\}$ are orthonormal to each other and $\|y\|_2 = 1$, then*

$$\sum_{i=1}^{n} \left(u_i^H y\right)^2 = 1$$

Proof. Set $U = (u_1, u_2, \ldots, u_n)$ then

$$\sum_{i=1}^{n} \left(u_i^H y\right)^2 = \left\|U^H y\right\|_2^2 = \|y\|_2^2 = 1$$

■

Theorem 18 *[13][p.19] Assume that $A \in \mathbb{C}^{m\times n}$ and $A = U\Sigma V^H$ be the* Singular Value Decomposition *of A with $\operatorname{rank}(A) = r \le n \le m$. Assume also that $\mathcal{M} = \{B : B \in \mathbb{C}^{m\times n} \text{ and } \operatorname{rank}(B) = k < r\}$ and $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^H$,with σ_i, u_i, v_i the singular values, left and right singular vectors of A, then*

$$\min_{B\in\mathcal{M}} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

Conclude also that the minimum 2-norm perturbation $\delta A \in \mathbb{C}^{m\times n}$ such that $\operatorname{rank}(A + \delta A) = k < r$ is $\delta A = -\sum_{i=k+1}^{r} \sigma_i u_i v_i^H$.

Proof. First, we prove

$$\|A - A_k\|_2 = \sigma_{k+1}$$

Since

$$U^H A_k V = \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$$

and

$$U^H AV = \text{diag}\,(\sigma_1, \ldots, \sigma_k, \ldots, \sigma_r, 0, \ldots, 0)$$

we have

$$\begin{aligned}
\|A - A_k\|_2 &= \|U^H (A - A_k) V\|_2 \\
&= \|\text{diag}\,(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_r, 0, \ldots, 0)\|_2 \\
&= \sigma_{k+1}
\end{aligned}$$

Clearly there exists $B \in \mathcal{M}$, namely A_k, such that $\min_{B \in \mathcal{M}} \|A - B\|_2 = \sigma_{k+1}$. It remains now to prove that $\|A - B\|_2 \geq \sigma_{k+1}$, $\forall B \in \mathcal{M}$. Since $k = \text{rank}\,(B)$ from lemma 15 we have $\text{null}\,(B) = n - k$. If we now choose $k+1$ of the n linearly independent columns of V, a dimension argument shows $\mathcal{N}(B) \cap \text{span}\,\{v_1, v_2, \ldots, v_{k+1}\} \neq \emptyset$. Consider now all the vectors y, with $\|y_2\| = 1$ that belong to the above intersection, then

$$\begin{aligned}
\|A - B\|_2^2 = \max_{\|y\|_2 = 1} \|(A - B)\, y\|_2^2 &\geq \|(A - B)\, y\|_2^2 = \|Ay\|_2^2 \\
&= \sum_{i=1}^{k+1} \sigma_i^2 \left(v_i^H y\right)^2 \|u_i\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 \left(v_i^H y\right)^2 \\
&\geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} \left(v_i^H y\right)^2 \overset{\text{(from lemma 17)}}{=} \sigma_{k+1}^2
\end{aligned}$$

Suppose now that $\delta A = B - A$, with $B \in \mathcal{M}$. The following two optimization problems are equivalent

$$\left\{ \min_{\delta A \in \mathbb{C}^{m \times n}} \|\delta A\|_2 : \text{rank}\,(A + \delta A) = k < r \right\} \Leftrightarrow \left\{ \min_{\delta A \in \mathbb{C}^{m \times n}} \|B - A\|_2 : \text{rank}\,(B) = k < r \right\}$$

Clearly the $B \in \mathcal{M}$ that satisfies the above is $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^H$, therefore

$$\delta A = \sum_{i=1}^{k} \sigma_i u_i v_i^H - \sum_{i=1}^{r} \sigma_i u_i v_i^H = - \sum_{i=k+1}^{r} \sigma_i u_i v_i^H$$

■

The above model would give us the minimum perturbation to a matrix to cause a rank deficiency. We will now state the major theorem which suggests a way to compute μ, in terms of the singular values.

Theorem 19 *Let $(B, A) \in \mathbb{R}^{n\times(m+n)}$ be controllable and $(\delta B, \delta A) \in \mathbb{C}^{n\times(m+n)}$ the minimum perturbation such that $(B + \delta B, A + \delta A)$ is uncontrollable. Then*

$$\mu = \|(\delta B, \delta A)\|_2 = \min_{\lambda\in\mathbb{C}} \sigma_n (B, A - \lambda I)$$

where $\sigma_n (B, A - \lambda I)$ is the smallest singular value of $(B, A - \lambda I)$.

Proof. This was first proven in [24] and then idependently in [10]. Other proofs have also apppeared, with the most appealing, in our view, in [17] and it is given here.
Since $(B + \delta B, A + \delta A)$ is uncontrollable, from condition (2) of theorem (13), we get

$$\text{rank}\,(B + \delta B, (A + \delta A) - \lambda I) < n, \text{ for some } \lambda \in \mathbb{C}$$

From theorem (18), the smallest perturbation that can make $(B, A - \lambda I)$ have rank less than n is $\sigma_n (B, A - \lambda I)$. Hence

$$\sigma_n (B, A - \lambda I) \leq \|(\delta B, \delta A)\|_2 \tag{1.4}$$

Clearly equality in (1.4) can be attained, by setting

$$(\delta B, \delta A) = -\sigma_n u_n v_n^H$$

where σ_n, u_n, v_n are the smallest singular value and the corresponding left and right singular vectors of $(B, A - \lambda I)$ for some $\lambda \in \mathbb{C}$. Taking the minimum over all $\lambda \in \mathbb{C}$ we get

$$\mu = \min_{(\delta B,\delta A)\in\mathbb{C}^{n\times(m+n)}} \|(\delta B, \delta A)\|_2 = \min_{\lambda\in\mathbb{C}} \sigma_n (B, A - \lambda I) \tag{1.5}$$

which clearly represent the minimum perturbation we are looking for. For future reference we will denote $\sigma(\lambda) \equiv \sigma_n (B, A - \lambda I) : \mathbb{C} \mapsto \mathbb{R}$. ■

This is a global optimization problem with respect to λ. Most approaches in literature consider a real system (A, B), but they allow complex perturbations $(\delta A, \delta B)$ such that $(A + \delta A, B + \delta B)$ is uncontrollable. The nice property of this minimum complex disturbances coincide with rank one matrices i.e both δA and δB are rank one matrices. The more realistic case where the perturbations $(\delta A, \delta B)$ are real, have been considered in [32], [10] and [24] . Solution strategies

for real perturbations considered in [10] resulted into an objective function which is discontinuous due to the formulation of the original problem. However the formulation considered in [32] turns to help avoid this discontinuity problem. A new version of Miminis' algorithm [24] called Density search algorithm [25], was also extended to real perturbations where a search is performed only on the real axis, which results into a real, λ, minimizing the objective function. The latter however, may not always compute the global minimum.

In this work, we will be implementing an algorithm that will minimize this function with respect to $\lambda \in \mathbb{C}$ as well as $\lambda \in \mathbb{R}$.

1.4 Organizational structure of Thesis

This thesis is organized as follows. In chapter (2) we review some papers on the subject. Specifically, we review the algorithm by Elsner and He [20], DeCarlo [32], Gu [14] and Burke, Lewis and Overton, [7]. In Chapter (3), we will Introduce the Density search algorithm [25] and the Tunneling algorithm [21] which we will be using for our algorithm implementation. Chapter (4) is devoted to the computation of the distance to uncontrollability (1.5) by combining Density search method and Tunneling method. In chapter (5), we will illustrate the performance of the algorithm in practice with some numerical examples. Performance of the algorithm will also be looked at when we consider $\lambda \in \mathbb{R}$. In chapter (6), we will give concluding remarks.

2 Review Of Existing Algorithms

2.1 Computing the Distance to Uncontrollability

The formulation (1.5) is a difficult problem to solve. It is a non-smooth global optimization problem in two variables, α and β, where $\lambda = \alpha + i\beta$. In the past, a number of algorithms have been designed to compute, μ both for $\lambda \in \mathbb{C}$ and $\lambda \in \mathbb{R}$. The algorithms designed to search for a local minimum cannot guarantee that they will compute the global minimum and the algorithms designed to search for the global minimum require computing time that is inversely proportional to $\mu^2(A, B)$ [4]. The cost is even excessively expensive in the case when the system is nearly uncontrollable. In this section, we briefly present two algorithms that compute the local minimum of the function (1.5) and also the two algorithms that compute the global minimum.

2.1.1 The Distance to Uncontrollability by Elsner and He

Since the problem of computing the distance to uncontrollability of, (1.5) is an optimization problem of minimizing, $\sigma(\lambda)$ over the complete complex plane, we present an algorithm by, Elsner and He's in [20] that compute the local minimum of this function. They turn to solve this by finding the zero points of the gradient of $\sigma(\lambda)$ using Newton's and the Bisection method.

Considering the case, $\lambda \in \mathbb{C}$, i.e., $\lambda = \alpha + i\beta$, this function $\sigma(\alpha + i\beta)$, becomes a real-valued function with a complex variable. This property makes the function, not analytic with respect to λ, however, the partial derivatives with respect to the real parameters α and β exist. This makes their method possible since the partial derivatives of $\sigma(\alpha, \beta)$, can be computed using the singular value decomposition of $(A - \lambda I, B)$.

Theory.

Let's consider the complex parameter, $\lambda = \alpha + i\beta$.

and let

$$(A - (\alpha + i\beta)I, B) = U(\alpha, \beta)\, \Sigma\, V^H(\alpha, \beta)$$

be the singular value decomposition of, $(A - (\alpha + i\beta)I, B)$ where Σ is the $n \times (m + n)$ rectangular matrix with non-negative real elements $\sigma_1 \geqslant \sigma_2 \geqslant, ..., \geqslant \sigma_n \geqslant 0$ at the diagonal, with $\sigma_n(\alpha, \beta) = \sigma(\alpha, \beta)$ being simple, $U(\alpha, \beta)$ is an $n \times n$ complex unitary matrix which is a function of α and β, whose columns are the left-singular vectors of $(A-(\alpha+i\beta)I, B)$ and $V(\alpha, \beta)$ is an $(m+n)\times(m+n)$ complex unitary matrix which is also a function of α and β, whose columns are the right-singular vectors of $(A - (\alpha + i\beta)I, B)$.

Also let's define a certain function, $s(\alpha, \beta) = s(\lambda)$ as,

$$s(\lambda) = v_n^H(\lambda) \begin{bmatrix} u_n(\lambda) \\ \\ 0 \end{bmatrix} \tag{2.1}$$

where u_n and v_n are the left and right singular vectors, with respect to the least singular value in the singular value decompositon of $(A - (\alpha + i\beta)I, B)$. Elsner and He in [20] showed that, the partial derivatives with respect to the real α and β can be defined and computed as

$$\frac{\partial \sigma(\alpha + i\beta)}{\partial \alpha} = -Re\, s(\alpha + i\beta) \tag{2.2}$$

$$\frac{\partial \sigma(\alpha + i\beta)}{\partial \beta} = -Im\, s(\alpha + i\beta) \tag{2.3}$$

Where "Re" and "Im" represent the real and imaginary components of the function $s(\lambda)$. Knowing the first derivatives, the second derivatives were easily calculated. The above equations (2.2) and (2.3) imply a relationship between the zero points of the function $s(\lambda)$ as defined in (2.1) and the critical points of $\sigma(\alpha, \beta)$, where a critical point of $\sigma(\alpha, \beta)$ is a point where the partial derivatives with respect to α and β are zero. From this nice relation it was shown in the theorem below in [20], that,

Theorem 20 *$\lambda^* = \alpha^* + i\beta^*$ is a zero of $s(\lambda)$ defined in (2.1), iff (α^*, β^*) is a critical point of $\sigma(\alpha, \beta) = \sigma_n(A - (\alpha + i\beta)I, B)$.*

Hence the computation of the distance to uncontrollability, μ is equivalent to finding the zero points of function (2.1), in which the real and imaginary parts $(Re\, s(\alpha, \beta), Im\, s(\alpha, \beta))$ will be the critical points of $\sigma(\alpha, \beta)$. Thus, some well-established root-finding methods, such as Newton's iterative algorithm or the Bisection method can be used to compute these critical points.

An interesting observation about the critical points is, they satisfy $\lambda^* = u_n^{*H} A u_n^*$, which means they lie in the field of values of A. Hence to decide which critical points are local minima, one can use the following well-known criterion. See for example [31][p.961] a critical point $\lambda^* = \alpha^* + i\beta^*$ of $\sigma(\lambda)$ is a local minimum of $\sigma(\alpha, \beta)$ if

$$(\frac{\partial^2 \sigma}{\partial \alpha^2})(\frac{\partial^2 \sigma}{\partial \beta^2}) - (\frac{\partial^2 \sigma}{\partial \alpha \partial \beta}) > 0$$

$$\frac{\partial^2 \sigma}{\partial \alpha^2} > 0$$

where $(\frac{\partial^2 \sigma}{\partial \alpha^2})$ and $(\frac{\partial^2 \sigma}{\partial \beta^2})$ represent the second derivatives with respect to the real and imaginary components, α and β. Newton's method needs a starting approximation. Since all critical points of λ satisfy $\lambda^* = u_n^{*H} A u_n^*$, all minimum points λ will lie in the field of values of A, hence according to Elsner and He in [20],

$$\lambda_{min}(\frac{A + A^T}{2}) \leqslant \alpha^* \leqslant \lambda_{max}(\frac{A + A^T}{2}), \quad \lambda_{min}(\frac{A - A^T}{2i}) \leqslant \beta^* \leqslant \lambda_{max}(\frac{A - A^T}{2i}) \tag{2.4}$$

Here $\lambda_{min}(A)$ and $\lambda_{max}(A)$ denote the minimal and the maximal eigenvalue of A. The function $\sigma(\lambda)$ is symmetric about the real axis (The proof of this symmetry is provided in chapter (3)), hence since $\sigma_n(A - \lambda^* I, B) = \sigma_n(A - \bar{\lambda}^* I, B)$, the search for minimum points can be restricted to

$$0 \leqslant \beta^* \leqslant \lambda_{max}(\frac{A - A^T}{2i})$$

Based on the above discussion, Newton's and Bisection methods presented in [20] can be used to to find μ for the cases of $\lambda \in \mathbb{C}$ and $\lambda \in \mathbb{R}$.

Complete proofs for these algorithms and convergence and also for the first and second derivatives can be found in [20].

Numerical Examples

Below we choose as an example, a system where we know that a complex λ gives its minimum.

$$A = \begin{bmatrix} 0.1419 & 0.7922 & 0.0357 \\ 0.4218 & 0.9595 & 0.8491 \\ 0.9157 & 0.6557 & 0.9340 \end{bmatrix}, B = \begin{bmatrix} 0.9572 \\ 0.4854 \\ 0.8003 \end{bmatrix}$$

In using Newton's method to compute μ, we run the algorithm by [20] for $\lambda \in \mathbb{C}$ on the above system and compared its results with the Density search algorithm [25] and a Trisection method in [23]. It gave a minimum of 0.3728 with a minimizer of $0.1260 + 0.3421i$ whiles Density search algorithm and the Trisection method [23] gave as minimums, 0.3710 and 0.3734 with minimizers as $0.1170 + 0.2814i$ and $0.1160 + 0.3471i$ respectively.

We also choose below, a system where we know a real λ gives its minimum.

$$A = \begin{bmatrix} 0.6787 & 0.3922 & 0.7060 \\ 0.7577 & 0.6555 & 0.0318 \\ 0.7431 & 0.1712 & 0.2769 \end{bmatrix}, B = \begin{bmatrix} 0.0462 \\ 0.0971 \\ 0.8235 \end{bmatrix}$$

Again in using Bisection method to compute μ, we run the algorithm by [20] for $\lambda \in \mathbb{R}$ on the above system and compared its results with the Density search algorithm for $\lambda \in \mathbb{R}$. It gave us a minimum 0.4176 with a minimizer of 0.2660 whiles Density search algorithm gave as a minimum, 0.3959 with a minimizer of 0.2460.

From the above numerical examples, it can be seen that, there is a general agreement in the minima attained by the three algorithms.The algorithm by [20] was able to locate a minimum which is almost close to the minimum identified by the Trisection method [23] and Density search method [25] for $\lambda \in \mathbb{C}$ and $\lambda \in \mathbb{R}$ respectively. Density search algorithm records the least function

value as compared to the other two. It was also observed that, the algorithm is able to converge quadratically when $\theta_k = 1$ is used in algorithm.

2.1.2 Distance to an Uncontrollable System by DeCarlo and Wicks

Newton's method used in [20] above, is based on the minimization of $\sigma_n(A-\lambda I, B)$ over all complex numbers λ. It requires a singular value decompositon computation at each iteration.

In this section, we state an algorithm due to Wicks and DeCarlo. The algorithm is iterative in nature without the need for searching or using a general minimization algorithm.

Mark Wick's and Decarlo's work in [32] suggest a new interpretation of the problem.

$$\mu = \min_{\lambda \in \mathbb{C}} \sigma_n \left(A - \lambda I, B\right) \tag{2.5}$$

as

$$\mu = \min_{q \in \mathbb{C}^n} \left\| [q^H A(I - qq^H), B] \right\|_2 \tag{2.6}$$

Based on this interpretation, they developed three algorithms for computing μ_R and μ_c, where μ_R and μ_c are the computed distances for $\lambda \in \mathbb{R}$ and $\lambda \in \mathbb{C}$ respectively.

Equation (2.6) is based on the observation that the pair (A,B) is uncontrollable if and only if there is a partitioning and a unitary coordinate transformation for which the equivalent transformed pair

$$\tilde{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tilde{B} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \tag{2.7}$$

has the submatrices A_{21} and B_2 equal to zero.

From this, they seek a sequence of orthonormal bases which successively decrease $||A_{12},\ B_2||$ where A_{21} and B_2 are as in equation (2.7). Hence the definition below will help in formalizing the problem.

Definition 21 *The distance measure, μ was then defined in [32] as;*

$$\mu^2 = \left\| e_n^H [A(I - e_n e_n^H), B] \right\|_2^2 = \sum_{j=1}^{n-1} |\alpha_{nj}|^2 + \sum_{j=1}^{m} |\beta_{nj}|^2 \tag{2.8}$$

Using the above definition, four algorithms were presented in [32], three of which are used when complex perturbations are allowed, $\lambda \in \mathbb{C}$. The fourth algorithm is used to compute λ in $\mathbb{R}$. Algorithm (1) presented below compute $\lambda \in \mathbb{C}$. All four complete algorithms and their proof of convergence can be found in [32].

Algorithm 1.

We also present a numerical example to demonstrate the working of algorithm (1). We are presenting a version of algorithm (1) in [32] which computes $\lambda \in \mathbb{C}$.

Algorithm 1 Case $\lambda \in \mathbb{C}$

Require: Given A and B

$\lambda_{new} = A(n, n)$

$U\,S\,V^H = \text{svd}(A - \lambda_{new} I, B)$

$\mu_{new} = S(n, n)$

repeat

$\lambda_{old} = \lambda_{new}$

$\mu_{old} = \mu_{new}$

$A = U^H A U$

$B = U^H B$

$\lambda_{new} = A(n, n)$

$U\,S\,V^H = \text{svd}(A - \lambda_{new} I, B)$

$\mu_{new} = S(n, n)$

until $\mu_{old} < \mu_{new}$

$\lambda_{min} = \lambda_{old}$

$\mu = \mu_{old}$

Upon termination, λ_{min} represent a critical point and μ represent the computed distance to un-

controllability for the given system.

Numerical Example

Consider the system below, where we know that a complex λ gives it's minimum;

$$A = \begin{bmatrix} 0.2259 & 0.9234 & 0.4389 & 0.2622 & 0.2967 \\ 0.1707 & 0.4302 & 0.1111 & 0.6028 & 0.3188 \\ 0.2277 & 0.1848 & 0.2581 & 0.7112 & 0.4242 \\ 0.4357 & 0.9049 & 0.4087 & 0.2217 & 0.5079 \\ 0.3111 & 0.9797 & 0.5949 & 0.1174 & 0.0855 \end{bmatrix}, B = \begin{bmatrix} 0.2625 \\ 0.8010 \\ 0.0292 \\ 0.9289 \\ 0.7303 \end{bmatrix}$$

Comparing the distance measure of the above system, using algorithm (1) in [32] above with the Density search algorithm [25], Algorithm (1) above converged to the minimum 0.1350 with a complex minimizer of $-0.2932 + 0.1052i$, whiles Density search algorithm gave as a minimum 0.1228, with a minimizer of $-0.3768 + 0.2260i$. Density search algorithm providing us with the smallest minimum.

The methods in [20] and [32] are the few methods that compute the local minimum. So in this work, we choose to present them briefly and test them on some numerical examples. Next we look at the other two methods which estimates the global minimum below.

2.1.3 Bisection And Trisection Algorithms

2.1.3.1 Gu's Verification Scheme

Ming Gu proposed the first algorithm scheme that accurately estimates the distance to uncontrollability in polynomial time, see [14]. This algorithm compares eigenvalues of matrix pencils involving kronecker products that depend on matrices A and B. Taking the computation of singular values and eigenvalues as atomic operations that can be performed in time cubic in the matrix dimension, Gu's test requires $O(n^6)$ operations.

Gu's test scheme is as follows, given two real numbers δ_1 and δ_2 with $\delta_1 > \delta_2 > 0$, the test returns either the information that

$$\mu \leqslant \delta_1 \tag{2.9}$$

or

$$\mu > \delta_2 \tag{2.10}$$

where, μ is the distance to uncontrollability measure of A and B. At least one of these statements must be true, even if both are true, only one of the two statements is verified.

Bisection Algorithm.

Based on this test, Gu used a bisection method to keep only an upper bound on the distance to uncontrollability. It refines the upper bound until condition (2.10) is satisfied. Complete verification scheme of equation (2.9) and (2.10) can be found in [14].

Sometimes it could be tempting to try to evaluate μ to higher precision by the bisection method. However, in order to make this work, one needs to set δ_1 and δ_2 sufficiently close to each other. Unfortunately, this lead to numerical difficulties, i.e., the necessary comparison of imaginary eigenvalues of the relevant pencils cannot be carried out with any confidence in the presence of rounding errors. Hence the Trisection variant algorithm below has the capability to improve upon that difficulty.

Trisection Method

The first improvement on [14] bisection method, was done by Burke, Lewis, and Overton in [7]. To obtain the distance to uncontrollability with better accuracy, Burke, Lewis and Overton, proposed a trisection variant step which replaces Gu's bisection step. The trisection algorithm bounds μ by an interval $[L, U]$ and reduces the length of this interval by a factor of $\frac{2}{3}$ at each iteration.

2.1.3.2 Gu's improved verification scheme by Mengi

Mengi, in his doctoral dissertation [23], presented an improvement to Gu's scheme that reduces the complexity from $O(n^6)$ to $O(n^4)$ on average and $O(n^5)$ in the worst case. In his modified scheme, A $2n^2 \times 2n^2$ generalized eigenvalue problem whose real eigenvalues are sought for in Gu's scheme are replaced by standard eigenvalue problem. With this improved verification scheme, Mengi used the trisection variant algorithm to compute the distance to uncontrollability. Refer to [23][14][7] for the complete form of this verification scheme and proofs.

Table (5.5) below compare numerical results between the Trisection algorithm in [23] and that of the Density search algorithm [25] which we will introduce in the next chapter. Specifically, we compare the global minimum, μ and the minimizers, λ attained by both algorithms for various matrices of different dimensions. For convenience we are representing the matrices by their "seed (Seeding the random number generator means initializing it to a certain status)", we first generate A and then B.

size(n, m)(seed)	Trisection		Density Search	
	μ	λ	μ	λ
(5,1). (1423).	0.1113	-0.5812 + 0.0161i	0.1088	-0.5683
(10,1). (1423).	0.0077	0.2976 + 0.0093i	0.0059	0.3004
(10,3). (1423).	0.0522	-0.0973 + 0.0445i	0.0483	-0.0650
(15,1). (1423).	0.0098	-0.1453 + 0.0033i	0.0083	-0.1366
(15,4). (1423).	0.0694	-0.1044 + 0.0199i	0.0672	-0.0848
(17,1). (1423).	0.0119	-0.0909 + 0.0106i	0.0108	-0.0754
(17,2). (1423).	0.0231	-0.0518 + 0.0104i	0.0215	-0.0379
(19,1). (1423).	0.0001	0	0.0000	-0.0002
(15,5). (1423).	0.0696	-0.1064 + 0.0200i	0.0672	-0.0858
(21,1). (1423).	0.0120	0.5515 + 0.7080i	0.0105	0.5531 + 0.7005i
(25,1). (1423).	0.0083	-1.1041 + 0.0138i	0.0029	-1.0916
(25,3). (1423).	0.0880	0.6450 + 0.0737i	0.0841	0.6990
(27,1). (1423).	0.0075	-0.3395 + 0.0057i	0.0070	-0.3352

Table 2.1: Numerical results

From the above table, we notice the general agreement in the computed distance to uncontrollability between these algorithms even though, Density search algorithm gives the least function as the global minimum.

3 Density Search and Tunneling Algorithm

3.1 Density Search Algorithm

3.1.1 Theory

In this chapter we present the necessary theory for the development of the Density search algorithm.

3.1.1.1 Transformation

In matrix computations, it is at times advantageous to simplify the original problem by introducing zeros in the given matrices. The following two well known lemmas [25], and theorem will aid in this transformation.

Lemma 22 *Let $U \in \mathbb{C}^{n\times n}$ be a unitary matrix and $x \in \mathbb{C}^n$. Then*

$$\|Ux\|_2 = \|x\|_2$$

Proof. By the definition of the 2-norm we know

$$\|Ux\|_2^2 = (Ux)^H (Ux) = x^H U^H U x = x^H x = \|x\|_2^2$$

■

Lemma 23 *Let $U \in \mathbb{C}^{m\times m}$ be a unitary matrix and $A \in \mathbb{C}^{m\times n}$. Then*

$$\|UA\|_2 = \|A\|_2$$

Proof. By the definition of the matrix norm we know

$$\begin{aligned}\|UA\|_2^2 &= \max_{\|y\|_2=1} \|(UA)\, y\|_2^2 = \max_{\|y\|_2=1} y^H A^H U^H U A y = \max_{\|y\|_2=1} y^H A^H A y \\ &= \max_{\|y\|_2=1} \|Ay\|_2^2 = \|A\|_2^2\end{aligned}$$

■

In a similar manner it can be shown, for the unitary matrix $V \in \mathbb{C}^{n\times n}$, $\|AV\|_2^2 = \|A\|_2^2$.

Theorem 24 *Orthogonal transformations of* (B, A) *preserve* μ.

Proof. Let μ be the distance of (B, A) from the nearest uncontrollable system, then we have the following :

$$\begin{aligned}\mu &= \min\left\{\|\delta B, \delta A\|_2 : (B+\delta B, A+\delta A) \in \mathcal{U}\right\} \\ &= \min\left\{\left\|U^T(\delta B, \delta A)\begin{pmatrix} V & 0 \\ 0 & U\end{pmatrix}\right\|_2 : U^T\left[(B+\delta B), (A+\delta A)\right]\begin{pmatrix} V & 0 \\ 0 & U\end{pmatrix} \in \mathcal{U}\right\}\end{aligned}$$

This proves that μ is also the distance of the system $\left(U^TBV, U^TAU\right)$ from the nearest uncontrollable system. ■

There are various techniques that introduce zeros to a matrix, so that the matrix is easier to manipulate. Here we will choose Householder transformations [30][p.235] since they use orthogonal matrices that effectively change our original system (B, A) to $(B, A) \equiv (U^TBV, U^TAU)$. We will choose U and V so that the new (B, A) is in the following *Controllability Canonical Form:*

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,k-1} & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2,k-1} & A_{2k} \\ & A_{32} & \cdots & A_{3,k-1} & A_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & A_{k,k-1} & A_{kk} \end{pmatrix}, \; B = \begin{pmatrix} A_{10} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.1}$$

see for example, [24]. Since from Theorem 24 we know that orthogonal transformations have no effect on μ, clearly the transformed system has the same μ as the original. So we can compute the *distance to uncontrollability* of (3.1) instead. The number $k \leq n$ is called the *controllability index* of (B, A). Using the form (3.1) we develop another theorem that will give us a numerically reliable algorithm to determine if the system is uncontrollable or not.

Theorem 25 *If* $A_{i,i-1} = 0$ *for some* $i \in \{2 : k\}$ *then* (B, A) *is uncontrollable.*

Proof. Assume (B, A) is controllable. Then theorem 3 implies that $\exists F = (F_1, F_2, \ldots, F_k)$ such that

$$\lambda(A) \cap \lambda(A + BF) = \emptyset$$

Algebraically

$$A + BF = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,k-1} & A_{1k} \\ A_{21} & A_{22} & \cdots & A_{2,k-1} & A_{2k} \\ & A_{32} & \cdots & A_{3,k-1} & A_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & A_{k,k-1} & A_{kk} \end{pmatrix} + \begin{pmatrix} A_{10} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} (F_1, F_2, \ldots, F_k)$$

$$= \begin{pmatrix} A_{11} + A_{10}F_1 & A_{12} + A_{10}F_2 & \cdots & A_{1,k-1} + A_{10}F_{k-1} & A_{1k} + A_{10}F_k \\ A_{21} & A_{22} & \cdots & A_{2,k-1} & A_{2k} \\ & A_{32} & \cdots & A_{3,k-1} & A_{3k} \\ & & \ddots & \vdots & \vdots \\ & & & A_{k,k-1} & A_{kk} \end{pmatrix}$$

Given $A_{i,i-1} = 0$, for some $i \in \{2 : k\}$ we may have the following partition:

$$= \left(\begin{array}{ccc|cc} A_{11} + A_{10}F_1 & \cdots & & A_{1,k-1} + A_{10}F_{k-1} & A_{1k} + A_{10}F_k \\ A_{21} & \cdots & & A_{2,k-1} & A_{2k} \\ & & & \vdots & \vdots \\ \hline & & A_{i,i-1} = 0 & & \\ & & & \vdots & \vdots \\ & & & A_{k,k-1} & A_{kk} \end{array}\right) \equiv \left(\begin{array}{c|c} C_1 & C_2 \\ \hline 0 & C_3 \end{array}\right)$$

Since this is a block upper triangular matrix we have

$$\lambda(A) \cap \lambda(A + BF) \supseteq \lambda(C_3) \neq \emptyset$$

which is a contradiction. ■

In the next two sections, 3.1.1.2 and 3.1.1.3, we investigate some of the properties of the function (1.5) that will pave the way towards its efficient computation.

3.1.1.2 The Derivative of $\sigma(\lambda)$

We begin with a well know theorem (see for example [6][p.65]) on conditions sufficient for differentiability of a complex function at a point $z_0 = x_0 + iy_0$, with $(x_0, y_0) \in \mathbb{R}\times\mathbb{R}$.

Theorem 26 *Consider the complex function*

$$f(z) = u(x,y) + iv(x,y)$$

defined on a neighborhood $@z_0$ *of* $z_0 = x_0 + iy_0$*, with* $(x_0, y_0) \in \mathbb{R}\times\mathbb{R}$*. Suppose also that*

1. *The partial derivatives* $u_x(x,y)$*,* $u_y(x,y)$*,* $v_x(x,y)$*,* $v_y(x,y)$ *exist everywhere in* $@z_0$*;*

2. *The partial derivatives are continuous at* (x_0, y_0) *and satisfy the* Cauchy-Riemann equations

$$u_x(x_0,y_0) = v_y(x_0,y_0),\ u_y(x_0,y_0) = -v_x(x_0,y_0) \tag{3.2}$$

Then $f'(z_0)$ *exists and satisfies*

$$f'(z_0) = u_x(x_0,y_0) + iv_x(x_0,y_0) \tag{3.3}$$

Corollary 27 *A real valued complex function* $f(z) : \mathbb{C} \mapsto \mathbb{R}$*,*

$$f(z) = u(x,y) + \underbrace{iv(x,y)}_{=0}$$

that satisfies the Caushy-Riemann equations at every point of a region $\mathcal{D}$ *is constant on* $\mathcal{D}$*, formally* $\forall z \in \mathcal{D} \Rightarrow f(z) = c \in \mathbb{R}$*. This implies that any not constant function* $f(z) : \mathbb{C} \mapsto \mathbb{R}$ *is not differentiable.*

Proof. With $x_0 + iy_0 \in \mathcal{D}$ we have

$$v_x(x_0,y_0) = v_y(x_0,y_0) = 0$$

from the Caushy-Riemann equations (3.2) we also have

$$u_x(x_0,y_0) = u_y(x_0,y_0) = 0$$

Therefore from (3.3) we have

$$f'(z) = 0, \forall z \in \mathcal{D}$$

meaning that $f(z)$ is constant on $\mathcal{D}$. ■

Remark 28 *Function $\sigma(\lambda) : \mathbb{C} \mapsto \mathbb{R}$ is clearly a real valued complex function*

$$\sigma(\lambda) = u(\alpha, \beta) + i(0), \ \lambda = \alpha + i\beta$$

Since $\sigma(\lambda)$ is not constant on $\mathbb{C}$, it is not differentiable on the complex plane, according to the last corollary. This means that some well known optimization techniques that find local optima cannot be used for the minimization of $\sigma(\lambda)$, on $\mathbb{C}$.

Turning now to linear algebra tools; from the singular value decomposition of $(B, A - \lambda I)$, with $\lambda \in \mathbb{C}$ we have

$$\Sigma(\lambda) \equiv \left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_n & \end{array}\right) = U^H(\lambda)(B, A - \lambda I) V(\lambda) \Longrightarrow$$

$$\left\{\begin{array}{l} \sigma_i(\lambda) = u_i^H(\lambda)(B, A - \lambda I) v_i(\lambda) \\ u_i(\lambda)\sigma_i(\lambda) = (B, A - \lambda I) v_i(\lambda) \\ \sigma_i(\lambda) v_i^H(\lambda) = u_i(\lambda)(B, A - \lambda I) \end{array}\right\}, i = 1 : n \tag{3.4}$$

The above and the following lemma will provide what is needed to prove theorem 32 below, regarding the derivative of $\dot{\sigma}_i(\lambda)$.

Lemma 29 *Given $x(t) = [\xi_i(t)] \in \mathbb{C}^n$ and $y(t) = [\psi_i(t)] \in \mathbb{C}^n$ with $t \in \mathbb{C}$ we have*

$$\left[y^H(t) x(t)\right]' = \dot{y}^H(t) x(t) + y^H(t) \dot{x}(t)$$

If also $W(t) = \left(\begin{array}{c} w_1^H(t) \\ \vdots \\ w_m^H(t) \end{array}\right) \in \mathbb{C}^{m \times n}$ *is partitioned by rows, we have*

$$[W(t) x(t)]' = \dot{W}(t) x(t) + W(t) \dot{x}(t)$$

Proof.

$$\begin{aligned}\left[y^H(t)\,x(t)\right]' &= \left[\sum_{i=1}^{n}\bar{\psi}_i(t)\,\xi_i(t)\right]' = \sum_{i=1}^{n}\left[\bar{\psi}_i(t)\,\xi_i(t)\right]' \\ &= \sum_{i=1}^{n}\bar{\psi}_i'(t)\,\xi_i(t) + \sum_{i=1}^{n}\bar{\psi}_i(t)\,\xi_i'(t) \\ &= \dot{y}^H(t)\,x(t) + y^H(t)\,\dot{x}(t) \end{aligned} \tag{3.5}$$

Also

$$W(t)\,x(t) = \begin{pmatrix} w_1^H(t)\,x(t) \\ \vdots \\ w_m^H(t)\,x(t) \end{pmatrix} \Longrightarrow$$

$$\left[W(t)\,x(t)\right]' = \begin{pmatrix} \dot{w}_1^H(t)\,x(t) \\ \vdots \\ \dot{w}_m^H(t)\,x(t) \end{pmatrix} + \begin{pmatrix} w_1^H(t)\,\dot{x}(t) \\ \vdots \\ w_m^H(t)\,\dot{x}(t) \end{pmatrix}$$

$$= \dot{W}(t)\,x(t) + W(t)\,\dot{x}(t)$$

■

Corollary 30 *If $y(t) = x(t)$ the derivative $\left[x(t)^H x(t)\right]'$ exists only at $x = 0$. In the real case however, $x(t) \in \mathbb{R}^n$ with $t \in \mathbb{R}$ the derivative does exist and satisfies*

$$\left[x^T(t)\,x(t)\right]' = 2\dot{x}^T(t)\,x(t)$$

Proof. When $y(t) = x(t)$ we have

$$\left[x^H(t)\,x(t)\right]' = \left[\sum_{i=1}^{n}\bar{\xi}_i(t)\,\xi_i(t)\right]' = \sum_{i=1}^{n}\left[\bar{\xi}_i(t)\,\xi_i(t)\right]' \tag{3.6}$$

In general, assume $\xi = \alpha + i\beta$ and set $\zeta = \bar{\xi}\xi = (\alpha^2 - \beta^2) + i\,(0)$. Then from the Cauchy-Riemann equations for ζ we have

$$u(\alpha,\beta) = \alpha^2 - \beta^2 \text{ and } v(\alpha,\beta) = 0$$

which gives $u_\alpha = 2\alpha$, $u_\beta = -2\beta$ and $v_\alpha = v_\beta = 0$. That is, the Cauchy-Riemann equations (3.2) hold only when $\xi = 0$. Therefore, from (3.6) $\left[x(t)^H x(t)\right]'$ exists only when $x = 0$.

When x is real however, we have from (3.5)

$$\begin{aligned}\left[x^T(t)\,x(t)\right]' &= \dot{x}^T(t)\,x(t) + x^T(t)\,\dot{x}(t)\\ &= 2\dot{x}^T(t)\,x(t)\end{aligned} \tag{3.7}$$

■

Remark 31 *If $x^T(t)\,x(t)$ is constant, from (3.7) clearly $\dot{x}^T(t)\,x(t) = 0$.*

Theorem 32 *The function $\sigma_i(\lambda) = \sigma_i(B, A-\lambda I)$, with $\lambda \in \mathbb{R}$ satisfies*

$$\dot{\sigma}_i(\lambda) = -u_i^T(\lambda)\,(0, I)\,v_i(\lambda)\,,\ i = 1:n$$

Proof. Using lemma 29 we may differentiate the first of (3.4) and get

$$\dot{\sigma}_i(\lambda) = \dot{u_i}^T(\lambda)\,(B, A-\lambda I)\,v_i(\lambda) + u_i^T(\lambda)\,(B, A-\lambda I)'\,v_i(\lambda) + u_i^T(\lambda)\,(B, A-\lambda I)\,\dot{v}_i(\lambda)$$

Using the second and third of (3.4) and that $(B, A-\lambda I)' = (0, -I)$ we get

$$\dot{\sigma}_i(\lambda) = \dot{u_i}^T(\lambda)\,u_i(\lambda)\,\sigma_i(\lambda) - u_i^T(\lambda)\,(0, I)\,v_i(\lambda) + \sigma_i(\lambda)\,v_i^T(\lambda)\,\dot{v}_i(\lambda) \tag{3.8}$$

Since $u_i^T(\lambda)\,u_i(\lambda) = v_i^T(\lambda)\,v_i(\lambda) = 1$, from remark 31 we have that

$$\dot{u_i}^T(\lambda)\,u_i(\lambda) = v_i^T(\lambda)\,\dot{v}_i(\lambda) = 0$$

and (3.8) becomes

$$\dot{\sigma}_i(\lambda) = -u_i^T(\lambda)\,(0, I)\,v_i(\lambda)\,,\ i = 1:n,\ \text{with } \lambda \in \mathbb{R}$$

Of course we will concentrate on the derivative of the smallest singular value of $(B, A-\lambda I)$, namely

$$\dot{\sigma}(\lambda) = -u_n^T(\lambda)\,(0, I)\,v_n(\lambda) \tag{3.9}$$

■

3.1.1.3 Bounds of $\sigma(\lambda)$

In this section we find bounds on the value of λ that minimizes $\sigma(\lambda)$. We will use the following lemma from [30][p.321]:

Lemma 33 *Assume $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ are the singular values of $A \in \mathbb{C}^{m\times n}$ and let $\tilde{\sigma_1} \geq \tilde{\sigma_2} \geq \cdots \geq \tilde{\sigma_n} > 0$ be the singular values of $B \in \mathbb{C}^{m\times n}$. Then*

$$|\sigma_i - \tilde{\sigma_i}| \leq \|A - B\|_2, \text{ with } i = 1:n$$

The following theorem uses the above lemma in order to find bounds for the value of $\lambda \in \mathbb{C}$ that minimizes $\sigma(\lambda)$.

Theorem 34 *Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ be the singular values of (B, A) and $\tilde{\lambda}$ the value of $\lambda \in \mathbb{C}$ that minimizes $\min_{\lambda\in\mathbb{C}} \sigma_n(B, A - \lambda I)$, that is*

$$\sigma\left(\tilde{\lambda}\right) \equiv \mu = \min_{\lambda\in\mathbb{C}} \sigma_n(B, A - \lambda I)$$

Then

$$\sigma_n - \tilde{\mu} \leq \left|\tilde{\lambda}\right| \leq \sigma_1 + \tilde{\mu}$$

where $\tilde{\mu} = \min_i \|A_{i,i-1}\|_2$, and $A_{i,i-1}$, with $i = 2:k$ being defined in (3.1).

Proof. Let $\sigma_1(\lambda) \geq \sigma_2(\lambda) \geq \cdots \geq \sigma_n(\lambda) > 0$ be the singular values of $(B, A - \lambda I)$. By lemma 33 we have

$$|\sigma(\lambda) - \sigma_n| \leq \|(B, A - \lambda I) - (B, A)\|_2 = \|(0, -\lambda I)\|_2 = |\lambda|$$

from which we get

$$|\sigma(\lambda) - \sigma_n| \leq |\lambda| \Leftrightarrow$$

$$\sigma_n - |\lambda| \leq \sigma(\lambda) \leq \sigma_n + |\lambda|$$

and for $\lambda = \tilde{\lambda}$

$$\sigma_n - \left|\tilde{\lambda}\right| \leq \mu \leq \sigma_n + \left|\tilde{\lambda}\right| \tag{3.10}$$

Also

$$|\sigma(\lambda) - |\lambda|| \leq \|(B, A - \lambda I) - (0, -\lambda I)\|_2 = \|(B, A)\|_2 = \sigma_1$$

from which we get

$$|\sigma(\lambda) - |\lambda|| \leq \sigma_1 \Leftrightarrow$$

$$|\lambda| - \sigma_1 \leq \sigma(\lambda) \leq |\lambda| + \sigma_1$$

and for $\lambda = \tilde{\lambda}$

$$\left|\tilde{\lambda}\right| - \sigma_1 \leq \mu \leq \left|\tilde{\lambda}\right| + \sigma_1 \tag{3.11}$$

We know also that a perturbation of the magnitude

$$\tilde{\mu} \equiv \min_{1 \leq i \leq k} \|A_{i,i-1}\|_2$$

at position $(i, i-1)$ of equation (3.1) will result into an uncontrollable system. Clearly though $\tilde{\mu}$ may not necessarily be the smallest such perturbation, therefore

$$\mu \leq \tilde{\mu}$$

From the left part of (3.11) now we get for $\left|\tilde{\lambda}\right|$

$$\left|\tilde{\lambda}\right| - \sigma_1 \leq \mu \leq \tilde{\mu} \Rightarrow \left|\tilde{\lambda}\right| \leq \sigma_1 + \tilde{\mu}$$

Similarly, from the left part of (3.10) we get for $\left|\tilde{\lambda}\right|$

$$\sigma_n - \left|\tilde{\lambda}\right| \leq \mu \leq \tilde{\mu} \Rightarrow \ \sigma_n - \tilde{\mu} \leq \left|\tilde{\lambda}\right|$$

Eventually

$$\sigma_n - \tilde{\mu} \leq \left|\tilde{\lambda}\right| \leq \sigma_1 + \tilde{\mu}$$

Graphically, $\tilde{\lambda}$ is within the following ring

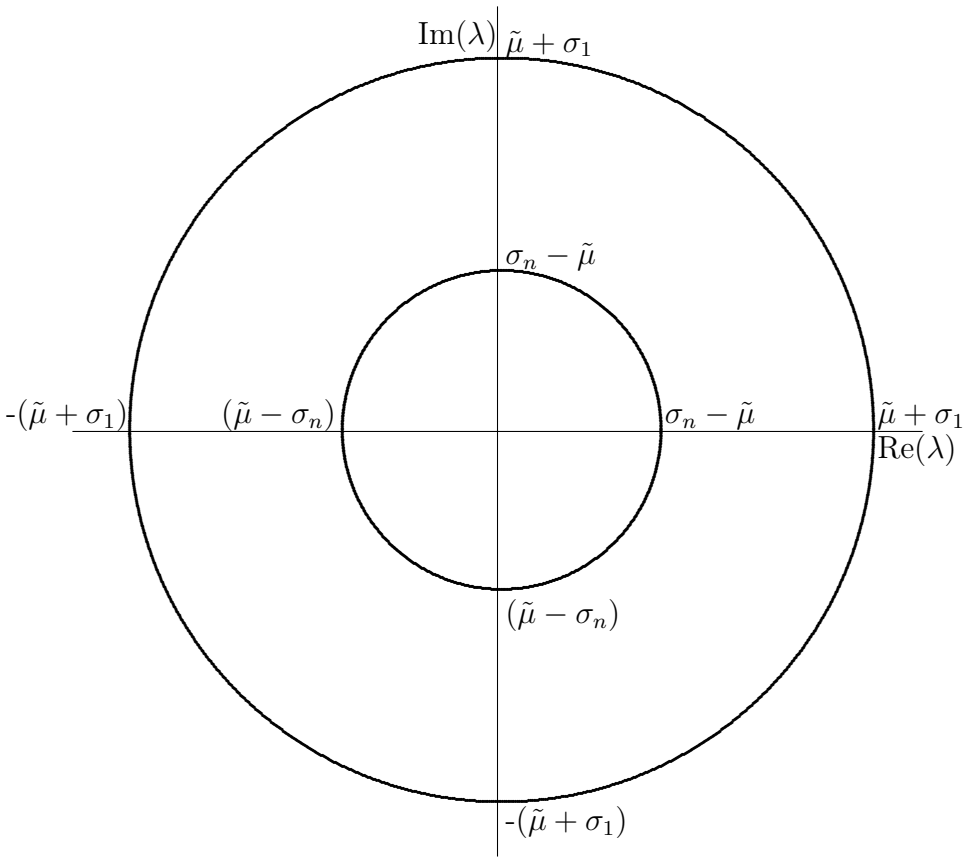

Figure 3.1 : The ring containing the minimum.

However, in the algorithms we are about to develop we will not make any use of the lower bound $\sigma_n - \tilde{\mu}$, since it can be negative. Furthermore even if it is not negative, the benefit from using it may not justify the effort of implementing it. Thus the above ring becomes a disc of radius $\sigma_1 + \tilde{\mu}$. There is also another convenient factor in the optimization of $\sigma(\lambda)$; the fact that it is symmetric about the real axis. Therefore we need only search one half of the disc defined by $\left|\tilde{\lambda}\right| \leq \sigma_1 + \tilde{\mu}$. The proof of this symmetricity follows, [24].

Theorem 35 *The function $\sigma(\lambda)$ is symmetric about the real axis.*

Proof. It is adequate to prove

$$\sigma_n(B, A - \lambda I) = \sigma_n\left(B, A - \bar{\lambda} I\right), \text{ for } \forall \lambda \in \mathbb{C}$$

From [30][p.267] we know that $\lambda(M) = \lambda\left(M^T\right)$, for $M \in \mathbb{R}^{n\times n}$. Since the squares of the singular values of the matrices $(B, A - \lambda I)$ and $\left(B, A - \bar{\lambda} I\right)$ are the eigenvalues of the matrices

$(B, A-\lambda I)^H (B, A-\lambda I)$ and $\left(B, A-\bar{\lambda} I\right)^H \left(B, A-\bar{\lambda} I\right)$ respectively, and that:

$$\begin{aligned}\left[(B, A-\lambda I)^H (B, A-\lambda I)\right]^T &= (B, A-\lambda I)^T \left[(B, A-\lambda I)^H\right]^T \\ &= (B, A-\lambda I)^T \left(B, A-\bar{\lambda} I\right) \\ &= \left(B, A-\bar{\lambda} I\right)^H \left(B, A-\bar{\lambda} I\right)\end{aligned}$$

since A and B are real, then

$$\sigma_i (B, A-\lambda I) = \sigma_i \left(B, A-\bar{\lambda} I\right),\ i = 1:n$$

QED. ■

3.1.1.4 Relative distance μ_r

In the way μ has been defined, it gives the absolute distance of a given system from the nearest uncontrollable system. However this may not be useful if one, for example, needs to compare the controllability of two systems. For instance, if μ_1 and μ_2 are the distances of (B_1, A_1) and (B_2, A_2) form their nearest uncontrollable systems respectively and $\mu_1 < \mu_2$ then (B_1, A_1) is not necessarily less controllable that (B_2, A_2). Actually, it may very well be the opposite if $\|(B_1, A_1)\|_2 \ll \|(B_2, A_2)\|_2$. So what we need, is to introduce the concept of the relative distance of a system from the nearest uncontrollable one.

Definition 36 *We define the relative distance of* (B, A) *from the nearest uncontrollable system* $(B+\delta B, A+\delta A)$ *as*

$$\mu_r = \|(B, A)\|_2^{-1} \min_{(\delta B, \delta A) \in \mathbb{C}^{n \times (m+n)}} \{\|(\delta B, \delta A)\|_2 : (B+\delta B, A+\delta A) \in \mathcal{U}\}$$

Clearly $\mu_r = \mu / \|(B, A)\|_2$.

In order to see how this definition will alter our computations note that if $D = USV^H$ is the singular value decomposition of a matrix D and α is a scalar then $\alpha D = U(\alpha S) V^H$. If we set

$\alpha = \|(B,A)\|_2^{-1}$ we get

$$\begin{aligned}\mu_r = \alpha\ \mu &= \alpha\ \min_{\lambda\in\mathbb{C}} \sigma_n\left(B, A-\lambda I\right)\\ &= \min_{\lambda\in\mathbb{C}} \sigma_n\left(\alpha B, \alpha A - \alpha\lambda I\right)\\ &= \min_{\hat{\lambda}\in\mathbb{C}} \sigma_n\left(\alpha B, \alpha A - \hat{\lambda} I\right), \text{ where } \hat{\lambda} = \alpha\lambda\end{aligned}$$

Furthermore,

$$\sigma_1\left(\alpha B, \alpha A\right) = \alpha\sigma_1\left(B,A\right) = 1$$

and since form (3.1) of $(\alpha B, \alpha A)$ we have

$$\alpha\left(B,A\right) = \alpha\left(A_{ij}\right)$$

define

$$\tilde{\mu}_r = \alpha\tilde{\mu}$$

In the sequel, we will work with the scaled system $(\alpha B, \alpha A)$ and we will compute μ_r instead. For economy of notation we will use

$$(\lambda, \mu, \tilde{\mu}) \equiv \left(\hat{\lambda}, \mu_r, \tilde{\mu}_r\right),\ (B,A) \equiv (\alpha B, \alpha A)$$

Note also that the relation $\left|\tilde{\lambda}\right| \leq \sigma_1 + \tilde{\mu}$ for the unscaled system, becomes $1 + \tilde{\mu}$ for the scaled system and we will denote it by η. We will also denote by $\mathcal{C}\left(0,\eta\right)$ the top half of the disc, with center at the origin and radius η. Below we introduce this algorithm for both instances of λ in $\mathbb{R}$ and $\mathbb{C}$.

3.1.2 Optimization in the interval $[-\eta, \eta]$

In this section we will solve the following optimization problem

$$\min_{\lambda\in[-\eta,\eta]} \sigma\left(\lambda\right) = \min_{\lambda\in[-\eta,\eta]} \sigma_n\left(B, A-\lambda I\right) \tag{3.12}$$

In general $\sigma\left(\lambda\right)$ will have more than one minima on the interval $[-\eta,\eta]$. Hence the above is a

global optimization problem. Consequently standard techniques like Newton's optimization cannot be applied unless the global minimum has been located with some certainty. The *density search algorithm* is an algorithm developed for computing the global minimum of a function (see [25]) and we will apply it on (3.12). Starting at $\lambda_1 = -\eta$ we will use

$$\lambda_{k+1} = \lambda_k + w_k$$

to locate the global minimum. The step, w_k will be computed so that the following two criteria are satisfied:

- The closer we get to a local minimum the denser the search becomes.
- The search should be complete in s steps, assuming $\sigma(\lambda) = \sigma$ for $\forall\lambda \in [-\eta, \eta]$, where σ is the expected value of $\sigma(\lambda)$ on $[-\eta, \eta]$.

The first property is clearly handled by

$$w_k = \sigma(\lambda_k)$$

that is, as the function becomes smaller the step size decreases. The second property will be met when

$$w_k = \frac{2\eta}{s} \tag{3.13}$$

Combining the effects of both into w_k we get

$$w_k = \sigma(\lambda_k)\frac{2\eta}{x} \tag{3.14}$$

where x is such that, the number of steps needed for locating the global minimum is s when $\sigma(\lambda) = \sigma$ for $\forall\lambda \in [-\eta, \eta]$, where σ is the expected value of $\sigma(\lambda)$ in $[-\eta, \eta]$. In this case

$$w_k = \sigma\frac{2\eta}{x} \tag{3.15}$$

Equating (3.13) and (3.15) we get

$$x = \sigma s$$

Eventually from (3.14) we get

$$w_k = \frac{\sigma(\lambda_k)}{\sigma}\frac{2\eta}{s} \tag{3.16}$$

The above model will have as a result a denser search when local minima are encountered, as well as a total number of steps approximating s when $\sigma(\lambda)$ is close to its expected value σ on $[-\eta, \eta]$. We are ready now to make the following remarks:

Remark 37 *Clearly the greater the value of s is, the better our estimation of the global minimum will be. We would prefer however an algorithm based on (3.16) to be as fast as possible. In view of this, we may first give s a "reasonable" value so that the computed* (λ, μ) *is a good approximations of the correct* (λ, μ)*. Then using the approximation of* (λ, μ) *and any standard optimization technique compute the correct value of* (λ, μ) *to machine precision. The latter is possible because the derivatives of* $\sigma(\lambda)$ *exist on* $\mathbb{R}$ *and we have already found a way to compute them.*

Remark 38 *In the extreme, but realistic cases, where the ratio* $\sigma(\lambda_k)/\sigma$ *is very large or very small we will encounter problems.*

1. *In the case where the ratio* $\sigma(\lambda_k)/\sigma$ *is very large,* w_k *will be very large as well and the danger of skipping some useful local minimum is very real. Very large values of* $\sigma(\lambda_k)/\sigma$ *are not useful for our computation so we will choose to ignore values that exceed some safe tolerance* `tolmax` *by setting them equal to* `tolmax`*.*

2. *In the case where the ratio* $\sigma(\lambda_k)/\sigma$ *is very small,* w_k *will be very small as well and the danger of iterating endlessly is real. Consider for example the case of an uncontrollable system where* $\mu = 0$*. Here too we should not allow* $\sigma(\lambda_k)/\sigma$ *to become smaller than a specific tolerance* `tol`*. If it does, we stop the computation declaring the system uncontrollable to machine precision.*

We are left now with the computation of σ. To compute this, let Λ be a random variable uniformly

distributed in $[-\eta, \eta]$. Then its density function $f(\lambda)$ satisfies

$$f(\lambda) = \begin{cases} \frac{1}{2\eta}, \text{ if } \lambda \in [-\eta, \eta] \\ 0, \text{ otherwise} \end{cases}$$

Thus

$$\sigma = \int_{-\eta}^{\eta} \sigma(\lambda) f(\lambda) d\lambda = \frac{1}{2\eta} \int_{-\eta}^{\eta} \sigma(\lambda) d\lambda$$

Using Simpson's rule (obviously there are many other choices here)

$$\begin{aligned} \sigma &= \frac{1}{2\eta} \left(\frac{\eta}{3} [\sigma(-\eta) + 4\sigma(0) + \sigma(\eta)] \right) \\ &= \frac{1}{6} [\sigma(-\eta) + 4\sigma(0) + \sigma(\eta)] \end{aligned}$$

Finally,

$$w_k = \frac{\sigma(\lambda_k)}{[\sigma(-\eta) + 4\sigma(0) + \sigma(\eta)]} \frac{12\eta}{s}$$

Starting at $-\eta$, and using w_k as a step size, the algorithm first estimates the global minimum by performing a systematic density search in $[-\eta, \eta]$. Then this minimum is used as an initial approximation to the root of the equation $\sigma'(\lambda)$.

In finding the root of this function $\sigma'(\lambda)$ with respect to the real λ, density search uses the secant method below.

Secant Method

This method is a root finding algorithm that uses a succession of roots of secant lines to better approximate a root of a function $\sigma'(\lambda)$. This method can be thought of as a finite difference approximation of Newton's method. It requires two initial values, λ_0 and λ_1, which should ideally be chosen to lie close to the root. Starting with initial values λ_0 and λ_1, we construct a line through the points $(\lambda_0, \sigma(\lambda_0))$ and $(\lambda_1, \sigma(\lambda_1))$ where $\sigma(\lambda_0)$ and $\sigma(\lambda_1)$, are the function values at points λ_0 and λ_1. We then find the root of this line and name it λ_2, etc, until convergence. This process can be modeled by the following iterative scheme:

$$\lambda_n = \lambda_{n-1} - \sigma(\lambda_{n-1}) \frac{\lambda_{n-1} - \lambda_{n-2}}{\sigma(\lambda_{n-1}) - \sigma(\lambda_{n-2})} \tag{3.17}$$

for $n = 2, 3, ...$ When this λ finally converges, the algorithm then computes the global minimum with respect to the minimizer.

Convergence of Secant Method

The iterate, λ_n, of the secant method converge to a root of σ', if the initial values λ_0 and λ_1 are sufficiently close to the root. The order of convergence is α, where $\alpha = \frac{1+\sqrt{5}}{2} \approx 1.618$

3.1.3 Optimization in the semi-disc, $\mathcal{C}(0, \eta)$

We have found that the minimum is inside a disc of radius η centered at the origin. Furthermore, the function is symmetric about the real axis and as such, reduces the search area to one half of the disc.

Let λ_{rc} be given by $\lambda_{rc} = \alpha_r + i\beta_{rc}$. We will perform a *two dimensional density search* following the same criteria as in the real case, where α_r will vary in the interval $[-\eta, \eta]$ with step w_k, while β_{rc} will vary along an axis parallel to the imaginary axis initiating at α_r (we term this the α_r axis) in the interval $\left[0, \sqrt{\eta^2 - \alpha_r^2}\right]$ with step q_k.

We start with $\lambda_{00} = \alpha_0 + i\beta_{00}$ where $\alpha_0 = -\eta$ and $\beta_{00} = 0$. Next we calculate $\sigma(\lambda_{00})$ and $w_0 = \sigma(\lambda_{00}) \frac{2\eta}{s\sigma}$, the step in the interval $[-\eta, \eta]$, where σ is the expected value of $\sigma(\lambda)$ in the upper half of the disc $(0, \eta)$ and s the number of steps in $[-\eta, \eta]$ if $\sigma(\lambda) = \sigma$ for $\forall\lambda \in [-\eta, \eta]$. Then compute $\lambda_{10} = \alpha_1 + i\beta_{10}$, where $\alpha_1 = \alpha_0 + w_0$ and $\beta_{10} = 0$. Now begins the search along the α_1 axis. First compute $\sigma(\lambda_{10})$ then let $q_0 = \sigma(\lambda_{10}) \frac{\sqrt{\eta^2 - \alpha_1^2}}{s_1\sigma}$ be the step along the α_1 axis, where s_1 is the number of steps we wish to have along the α_1 axis in the interval $\left[0, \sqrt{\eta^2 - \alpha_1^2}\right]$ if $\sigma(\lambda) = \sigma$ for $\forall\lambda \in \left[0, \sqrt{\eta^2 - \alpha_1^2}\right]$. Then compute $\lambda_{11} = \alpha_1 + i(\beta_{10} + q_0)$ and continue this line of search until $|\lambda_{1m}| > \eta$ for some m. Once the search on the α_1 axis is complete we are ready to move along the real axis once more. To do so we choose as step

$$w_1 = \min_j \sigma(\lambda_{1j}) \frac{2\eta}{s\sigma}$$

where $\min_j \sigma(\lambda_{1j})$ is the minimum value of $\sigma(\lambda)$ encountered during the search along the α_1 axis.

Set $\lambda_{20} = \alpha_2 + i\beta_{20}$ with $\alpha_2 = \alpha_1 + w_1$ and $\beta_{20} = 0$. Continue until $\alpha_n > \eta$ for some n.

We will now compute s_r for $r = 1 : n-1$, so that we have a search along the α_r axis that will work along the exact same principles as the search on the real axis. To do this we need to find a relation between s_r and α_r. First we observe that $s_r = 0$ when $\alpha_r = -\eta$ or $\alpha_r = \eta$, then we require that $s_r = \frac{s}{2}$ when $\alpha_r = 0$ and that s_r has the same value for α_r as for $-\alpha_r$, that is, the graph of s_r is symmetric about the s_r axis. The latter two requirements are meaningful under the assumption $\sigma(\lambda) = \sigma$ for $\forall \lambda \in \mathcal{C}(0, \eta)$.

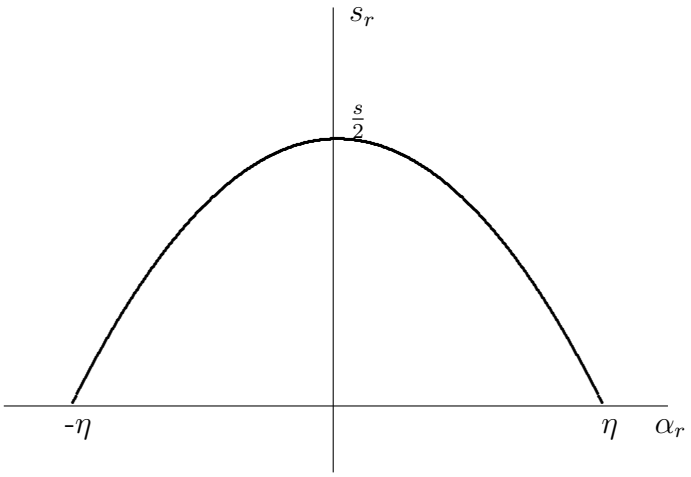

Figure 4.1 : A graphical representation of the step value dependent on the current real axis location.

According to our requirements the most suitable function of s_r in relation to α_r is quadratic, higher degree functions would introduce parameters that will give freedom that is not required. So we have

$$s_r = \varepsilon\alpha_r^2 + \delta\alpha_r + \gamma$$

along with the following conclusions: $\alpha_r = 0 \Rightarrow s_r = \frac{s}{2}$ gives $\gamma = \frac{s}{2}$; $s_r(\alpha_r) = s_r(-\alpha_r) \Rightarrow \delta = 0$; finally $\alpha_r = \eta \Rightarrow s_r = 0$ gives $\varepsilon = -\frac{s}{2\eta^2}$. Therefore

$$s_r = \frac{s}{2}\left(1 - \frac{\alpha_r^2}{\eta^2}\right)$$

Finally, we need to compute the expected value of $\sigma(\lambda)$ in $\mathcal{C}(0,\eta)$. Let Λ be a random variable equally distributed in $\mathcal{C}(0,\eta)$, and let $\lambda = \alpha + i\beta$ be a value of Λ, then the density function of Λ is

$$f(\alpha,\beta) = \begin{cases} \frac{2}{\pi\eta^2}, \text{ if } \alpha \in [-\eta,\eta] \text{ and } \beta \in \left[0, \sqrt{\eta^2-\alpha^2}\right] \\ 0 \text{ otherwise} \end{cases}$$

with expected value

$$\sigma = \int_0^{\sqrt{\eta^2-\alpha^2}} \int_{-\eta}^{\eta} \sigma(\alpha,\beta) f(\alpha,\beta)\, d\alpha\, d\beta, \text{ where } \sigma(\alpha,\beta) \equiv \sigma(\lambda)$$

Converting to polar coordinates we get

$$\alpha = \rho\cos\theta \text{ and } \beta = \rho\sin\theta, \text{ where } \rho \in [0,\eta] \text{ and } \theta \in [0,\pi]$$

The density function becomes

$$f(\rho,\theta) = \begin{cases} \frac{2}{\pi\eta^2}, \text{ if } \rho \in [0,\eta] \text{ and } \theta \in [0,\pi] \\ 0 \text{ otherwise} \end{cases}$$

and the expected value is

$$\sigma = \int_0^{\eta} \int_0^{\pi} \sigma(\rho,\theta) f(\rho,\theta) \left| \frac{\vartheta(\alpha,\beta)}{\vartheta(\rho,\theta)} \right| d\theta d\rho$$

where

$$\frac{\vartheta(\alpha,\beta)}{\vartheta(\rho,\theta)} = \left| \begin{pmatrix} -\rho\sin\theta & \cos\theta \\ \rho\cos\theta & \sin\theta \end{pmatrix} \right| = -\rho\sin^2\theta - \rho\cos^2\theta = -\rho$$

Therefore

$$\begin{aligned} \sigma &= \int_0^{\eta} \int_0^{\pi} \sigma(\rho,\theta) f(\rho,\theta)\, \rho d\theta d\rho \\ &= \frac{2}{\pi\eta^2} \int_0^{\eta} \rho \int_0^{\pi} \sigma(\rho,\theta)\, d\theta d\rho \end{aligned}$$

Using Simpson's rule (there are many choices here too) we get

$$\begin{aligned} \sigma &= \frac{2}{\pi\eta^2} \int_0^{\eta} \rho \left[\frac{\pi}{6} \left[\sigma(\rho,0) + 4\sigma\left(\rho, \frac{\pi}{2}\right) + \sigma(\rho,\pi) \right] \right] d\rho \\ &= \frac{1}{3\eta^2} \int_0^{\eta} \rho\sigma(\rho,0)\, d\rho + \frac{4}{3\eta^2} \int_0^{\eta} \rho\sigma\left(\rho, \frac{\pi}{2}\right) d\rho + \frac{1}{3\eta^2} \int_0^{\eta} \rho\sigma(\rho,\pi)\, d\rho \end{aligned}$$

Once again, using Simpson's rule for each of the above three integrals we get

$$\sigma = \frac{1}{3\eta^2}\left[\frac{\eta}{6}\left[0\sigma\left(0,0\right) + \frac{4\eta}{2}\sigma\left(\frac{\eta}{2},0\right) + \eta\sigma\left(\eta,0\right)\right]\right] +$$
$$\frac{4}{3\eta^2}\left[\frac{\eta}{6}\left[0\sigma\left(0,\frac{\pi}{2}\right) + \frac{4\eta}{2}\sigma\left(\frac{\eta}{2},\frac{\pi}{2}\right) + \eta\sigma\left(\eta,\frac{\pi}{2}\right)\right]\right] +$$
$$\frac{1}{3\eta^2}\left[\frac{\eta}{6}\left[0\sigma\left(0,\pi\right) + \frac{4\eta}{2}\sigma\left(\frac{\eta}{2},\pi\right) + \eta\sigma\left(\eta,\pi\right)\right]\right] \Longrightarrow$$

$$\sigma = \frac{1}{9}\left[\sigma\left(\frac{\eta}{2},0\right) + \frac{1}{2}\sigma\left(\eta,0\right)\right] + \frac{4}{9}\left[\sigma\left(\frac{\eta}{2},\frac{\pi}{2}\right) + \frac{1}{2}\sigma\left(\eta,\frac{\pi}{2}\right)\right] +$$
$$\frac{1}{9}\left[\sigma\left(\frac{\eta}{2},\pi\right) + \frac{1}{2}\sigma\left(\eta,\pi\right)\right]$$
$$= \frac{1}{9}\left[\sigma\left(\frac{\eta}{2},0\right) + 4\sigma\left(\frac{\eta}{2},\frac{\pi}{2}\right) + \sigma\left(\frac{\eta}{2},\pi\right)\right] + \frac{1}{18}\left[\sigma\left(\eta,0\right) + 4\sigma\left(\eta,\frac{\pi}{2}\right) + \sigma\left(\eta,\pi\right)\right]$$

Converting back to Cartesian coordinates we get

$$\sigma = \frac{1}{9}\left[\sigma\left(\frac{\eta}{2},0\right) + 4\sigma\left(0,\frac{\eta}{2}\right) + \sigma\left(-\frac{\eta}{2},0\right)\right] + \frac{1}{18}\left[\sigma\left(\eta,0\right) + 4\sigma\left(0,\eta\right) + \sigma\left(-\eta,0\right)\right] \tag{3.18}$$

Remark 39 *As with the search in $[-\eta,\eta]$, here too we can only get an approximation of the $\lambda \in \mathbb{C}$ that minimizes $\sigma(\lambda)$. This approximation becomes better as s increases. However, it is obvious that we would like to keep s as low as possible and still get a good approximation of $\lambda \in \mathbb{C}$ that minimizes $\sigma(\lambda)$. Since the derivatives of $\sigma(\lambda)$ do not exist anywhere in the complex plane, except at the origin, standard optimization methods can not be used to compute the optimum. There is however a method in [1] that successfully computes a local optimum of $\sigma(\lambda)$, converging linearly to the solution. This method has been implemented as part of the Density search method. It is worth pointing out that remark 38 holds here too.*

3.2 Tunneling Algorithm

In the move along the imaginary axis of the complex plane, and with an initial starting point from the step point on the real axis α_r, we will use the Tunneling algorithm technique to perform local searches to approach any local minimum.

The Tunnneling method attempts to solve the global optimization problem by performing local searches such that, at each time, a different local minimum is reached [21].

3.2.1 Basic structure of the Tunneling algorithm

The Tunneling algorithm is composed of sequence of cycles, each cycle consist of two phases, a minimization phase and a tunneling phase [21].

In the minimization phase, given a starting point, $\hat{\lambda}$, we use any minimization algorithm to find a local minimizer of $\sigma(\lambda)$, say λ^*. Any optimization algorithm with a descent property on $\sigma(\lambda)$ can be used to find the local minimum [9].

The second phase is called tunneling phase and the purpose is to obtain a good starting point, $\hat{\lambda}$, different from, λ^*, but with the same function value as, λ^*, for the next minimization phase. Starting from the point, λ^*, obtained in the previous minimization phase, we find the zero of a function called, the tunneling function;

$$T(\lambda) = \sigma(\lambda) - \sigma(\lambda^*). \tag{3.19}$$

Where $\sigma(\lambda^*)$ is the function value obtained during the previous minimization phase. Once the solution of equation (3.19) is obtained for a $\lambda \neq \lambda^*$, this point is taken as the starting point, $\hat{\lambda} = \lambda$, for the next minimization phase. If a zero of equation (3.19) cannot be found after a suffiecient computer time is used by some zero finding algorithm, then the whole algorithm is terminated. To decide when the algorithm is terminated is a subjective decision of the user. In our case, it is when $\lambda > \eta$ [9].

A Geometric interpretation of the Tunneling algorithm.

The alternate use of minimizations and tunneling phases in the Tunneling algorithm are illustrated geometrically in the figure (3.1) below.

Starting from an arbitrary point λ_1^0, a minimzation phase is performed, ending at the local minimum

at the λ_1^* with a function value of $\sigma(\lambda_1^*)$. In this phase, the function value decreases since

$$\sigma(\lambda_1^*) < \sigma(\lambda_1^0) \tag{3.20}$$

With the known values of λ_1^* and $\sigma(\lambda_1^*)$, a tunneling phase is started, ending at λ_2^0, which is the zero of the tunneling function, equation (3.19). This point λ_2^0, has the function value $\sigma(\lambda_2^0) = \sigma(\lambda_1^*)$. In this phase, the function value is not necessarily lowered but instead, an excellent starting point is obtained for the next minimization phase since there is a guarantee that, the next local minimum we find λ_2^*, will have a function value

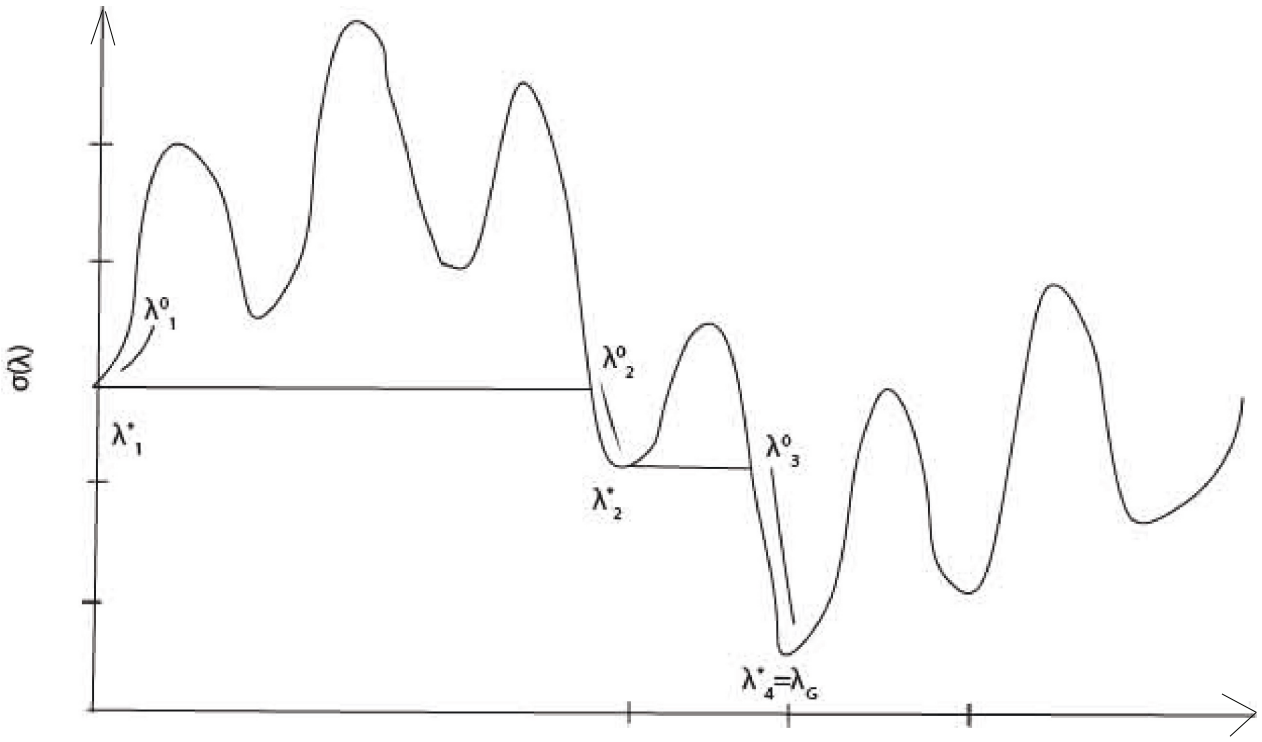

Figure 3.1: "Tunneling" of irrelevant minima.

$$\sigma(\lambda_2^*) \leqslant \sigma(\lambda_2^0) \tag{3.21}$$

and consequently from (3.20) and (3.21)

$$\sigma(\lambda_2^*) \leqslant \sigma(\lambda_1^*) \tag{3.22}$$

Once we have λ_2^*, we start a new tunneling phase, looking for a λ_3^0, the zero of equation (3.19) at the level of the last local minimum $\sigma(\lambda_2^*)$. At λ_3^0, the tunneling phase is ended, and a minimization phase, will take us from λ_3^0 to the "local" minimum at λ_4^*. The algorithm will start now the next tunneling phase looking for a zero of the tunneling function at the level $\sigma(\lambda_4^*)$. After a certain computer time, tmax, is spent without finding a zero of the tunneling function, the user can assume that "probably" $\lambda_G = \lambda_4^*$ is the global minimum [9].

Tunneling function additional parameters

The tunneling function, as given in equation (3.19), could be used with some zero finding algorithm during the tunneling phase, except for the fact that at the point λ^*, previous minimization phase found a local minimum that has become now also a zero of equation (3.19).

Since we want a zero of equation (3.19), say $\hat{\lambda}$, with $\hat{\lambda} \neq \lambda^*$, it is better to cancel or "deflate" the zero of equation (3.19) at λ^*, introducing a pole at λ^*, with a pole strength τ, obtaining a more suitable definition for the tunneling function given by

$$T(\lambda) = \frac{\sigma(\lambda) - \sigma(\lambda^*)}{(\prod_{i=1}^{l}(\lambda - \lambda_i^*)^{\tau_i})(\lambda - \lambda_m)^x} \tag{3.23}$$

The term $(\sigma(\lambda) - \sigma(\lambda^*))$ eliminates as possible solutions, all those points λ satisfying $\sigma(\lambda) > \sigma(\lambda^*)$. To prevent the algorithm from locating as solutions previous minimizers found at λ_i^*, $i = 1, 2, ..., l$, with function value $\sigma(\lambda_1^*) = \sigma(\lambda_2^*) = ... = \sigma(\lambda_i^*) = \sigma(\lambda^*)$, we place a pole, $(\lambda - \lambda^*)^\tau$ with pole strength τ at position λ^*. The term $(\lambda - \lambda_m)^x$ called, mobile pole is also added to smooth out, any irrelevant local minimizer that might attract any particular minimization algorithm during the search for $\hat{\lambda}$ and x is its strength [21].

At the begining of each tunneling phase, these parameters are updated automatically by the method. To compute x, starting from a value of $x = 0$, small increments, say $\tau = 1.0$, are added

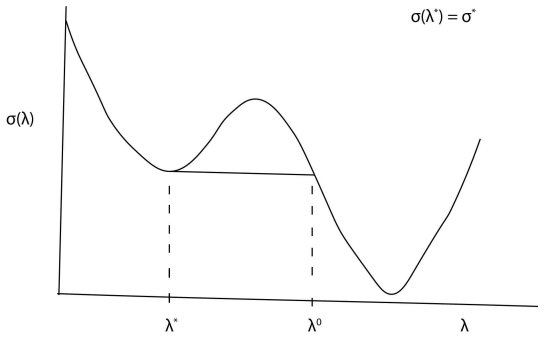

Figure 3.2: Poles placed at λ^*

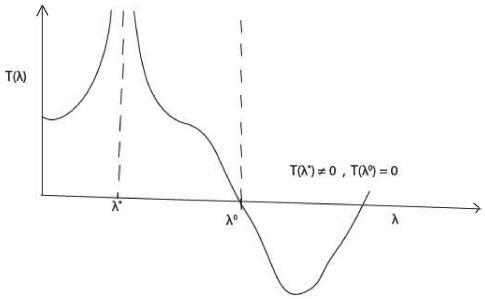

Figure 3.3: Transformed function after poles were placed at λ^* to deflate the zero of the tunneling function.

to τ until the pole is strong enough to cancel $T(\lambda^*)$. The geometric interpretation of what this parameter does is shown in figure (3.2) and figure (3.3) above.

General stopping conditions of the Tunneling algorithm

1. In the minimization phase, a convergence condition is set with tolerance, $\epsilon = 10^{-7}$.

2. In the Tunneling phase, the algorithm is stopped whenever the condition $|T(\lambda)| < \epsilon$ is satisfied.

3. This option occurs when for a given ϵ, in a given number of iterations, the zero finding algorithm fails to locate $\hat{\lambda}$. If this is the case, we are in the position of selecting a bigger ϵ, until the satisfaction of the inequality is achieved; this procedure could be repeated a specified number of times.

3.2.2 Algorithms used in both the Minimization and Tunneling Phases

Obviously there are many choices of optimization algorithms out there that can be used in both the minimization and tunneling phases. We will be using the gradient descent method in the minimization phase and the Newton's method in the tunneling phase in this algorithm.

3.2.2.1 Gradient descent method used in Minimization Phase

Gradient descent is a first-order iterative optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Description

Gradient descent is based on the observation that if the function $\sigma(\lambda)$ is defined, and differentiable in a neighborhood of a point a, then $\sigma(\lambda)$ decreases fastest if one goes from a in the direction of

the negative gradient of σ at a, $-\sigma'(a)$. It follows that if,

$$b = a - \gamma\sigma'(a)$$

where γ is the scaling factor and $\sigma'(a)$ is the derivative of the function, σ at point, a. For, γ small enough, then, $\sigma(a) \geqslant \sigma(b)$. In other words, the term $\gamma\sigma'(a)$ is subtracted from a because we want to move against the gradient, namely down towards the minimum. With this observation in mind, one starts with an initial guess of λ_0^* for a local minimum of function, σ, and considers the sequence $\lambda_0^*, \lambda_1^*, \lambda_2^*, ...$ such that

$$\lambda_{k+1}^* = \lambda_k^* - \gamma\sigma'(\lambda_k^*),$$

for $k = 1, 2, ..., n$ we have;

$$\sigma(\lambda_1^*) \geqslant \sigma(\lambda_2^*) \geqslant \sigma(\lambda_3^*) \geqslant, ..., \sigma(\lambda_n^*)$$

So hopefully the sequence (λ_n^*) converges to the desired local minimum. The value of the scaling factor γ is allowed to change at every iteration. With certain assumptions on the function $\sigma(\lambda^*)$ and particular choices of γ, convergence to a local minimum can be guaranteed.

3.2.2.2 Newton's Method in The Tunneling Phase

As described in the previous section, the tunneling phase requires a zero finding algorithm to obtain the solution of the tunneling function given by equation (3.19). Newton's method is a method for finding successively better approximations to the roots of this function. It is known that Newton's method will converge to these zero points if the starting point is within a neighborhood δ , of the solution.

$$\lambda : T(\lambda) = 0$$

With an initial guess point λ_0 for a root of the function T, the method uses the following iteration to converge to the solution;

$$\lambda_{n+1} = \lambda_n - \frac{T(\lambda_n)}{T'(\lambda_n)} \tag{3.24}$$

for $n = 0, 1, 2, ...$ until a sufficiently accurate value is reached.

4 The Two-Phase Algorithm

4.1 Algorithm Implementation

This algorithm would compute the distance of a linear dynamic system from the nearest uncontrollable system. The problem of computing this distance is equivalent to the global optimization problem;

$$\mu = \min_{\lambda \in \mathbb{C}} \sigma\left(B, A - \lambda I\right)$$

Before we proceed to introduce the Two-phase algorithm, it is worth pointing out that, we will initialize the system by reducing it into a simpler form. We will transform the original given system A and B into its *Controllability Canonical Form* (3.1). We will also compute the partial derivatives of $\sigma(\alpha + i\beta)$ with respect to β, using equation (2.3).

The Two-Phase Algorithm works like the Density Search Algorithm except that the search along the α_r axis is performed using the Tunneling Algorithm.

In the **minimization phase** of the Tunneling algorithm, we use the gradient descent method to locate a local minimum as in:

$$\beta_{k+1} = \beta_k - \gamma(Im\, s(\beta_k))$$

for $k = 1, 2, ...$, with $\beta_1 = \delta$. We continue with the iteration until the given convergent condition, $|\beta_{k+1} - \beta_k| \leqslant \epsilon$ in this phase is met. In our examples, we set the tolerance to be $\epsilon \leqslant 10^{-7}$, the scaling factor γ to be 0.1 and $\delta = 0.01$.

Next we switch to the **Tunneling phase**. The minimization-tunneling iteration is repeated until β_k for some k exceeds $\sqrt{\eta^2 - \alpha_r^2}$. We then set

$$\alpha_{r+1} = \alpha_r + w_r \quad for \quad r = 1, 2, ..., until\, \alpha_r > \eta$$

using;

$$w_r = \min_j \sigma\left(\lambda_{1j}\right) \frac{2\eta}{s\sigma}$$

as the step, where $\min_j \sigma\left(\lambda_{1j}\right)$ is the minimum value of $\sigma(\lambda)$ encountered during the search along the α_r axis.

The above algorithm was implemented in MATLAB and was used to test some chosen systems. Numerical results are shown in the next chapter.

5 Numerical Results

5.1 Numerical Examples

We run the Two-phase algorithm on systems with random coefficients of various sizes. The results of the Trisection method [23] and Density search algorithm [25] are considered to be accurate, hence we compared the results of this new method with these two algorithms.

Example (1). Consider the $A \in \mathbb{R}^{5\times5}$ and $B \in \mathbb{R}^{5\times1}$ system with a "seed" of [1338]

$$A = \begin{bmatrix} 0.6083 & 0.4087 & 0.1122 & 0.0055 & 0.6650 \\ 0.9759 & 0.2867 & 0.0417 & 0.4811 & 0.0768 \\ 0.0398 & 0.7915 & 0.7535 & 0.8346 & 0.4322 \\ 0.7477 & 0.7409 & 0.0994 & 0.1307 & 0.1439 \\ 0.3844 & 0.5452 & 0.0276 & 0.5774 & 0.0607 \end{bmatrix}, B = \begin{bmatrix} 0.6083 \\ 0.9759 \\ 0.0398 \\ 0.7477 \\ 0.3844 \end{bmatrix}$$

Method	Results	
	Global Min	λ
Trisection	0.0366	-0.4693 + 0.0085i
Density search	0.0347	-0.46319
Two-phase	0.0354	-0.4499

Table 5.1: Results for system 1

It can be seen that with respect to Density search and Two-phase algorithms, the minima are attained at the point of a *Real* λ, even though the search was done allowing $\lambda \in \mathbb{C}$. This shows that, it is possible to allow only complex perturbations, yet it is a real value of λ that would minimize the system. Also from observation, Density search algorithm records the least function

value as the global minimum. Below is a 3-D and a contour plot of the system. From the contour plots below, three local minima were encountered.

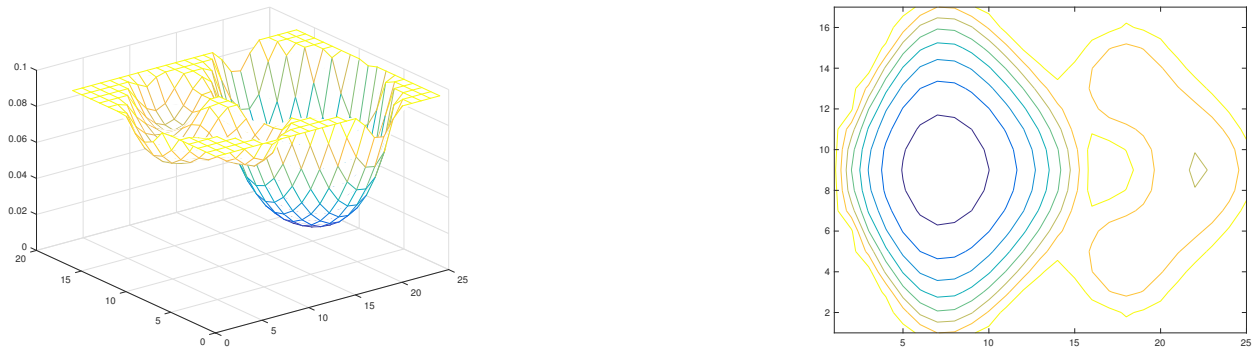

Figure 5.1: System 1 plots.

Example (2). Consider the $A \in \mathbb{R}^{5\times5}$ and $B \in \mathbb{R}^{5\times1}$ system with a "seed" of [1248]

$$A = \begin{bmatrix} 0.2963 & 0.7112 & 0.8771 & 0.9134 & 0.3004 \\ 0.5010 & 0.9892 & 0.0170 & 0.4260 & 0.9938 \\ 0.2073 & 0.5890 & 0.1159 & 0.9368 & 0.4511 \\ 0.2073 & 0.4238 & 0.6096 & 0.2610 & 0.0129 \\ 0.7714 & 0.1449 & 0.0508 & 0.0884 & 0.5913 \end{bmatrix}, B = \begin{bmatrix} 0.2963 \\ 0.5010 \\ 0.5788 \\ 0.2073 \\ 0.7714 \end{bmatrix}$$

Method	Results	
	Global Min	λ
Trisection	0.0452	-0.5486 + 0.0087i
Density search	0.0445	-0.5638
Two-phase	0.0460	-0.5719

Table 5.2: Results for system 2

Trisection method records a complex minimizer whiles Density search and Two-phase records a

real minimizer. In this example also, the least function value was attained by the Density search method. Below is a 3-D and a contour plot of the system. Three local minima were encountered.

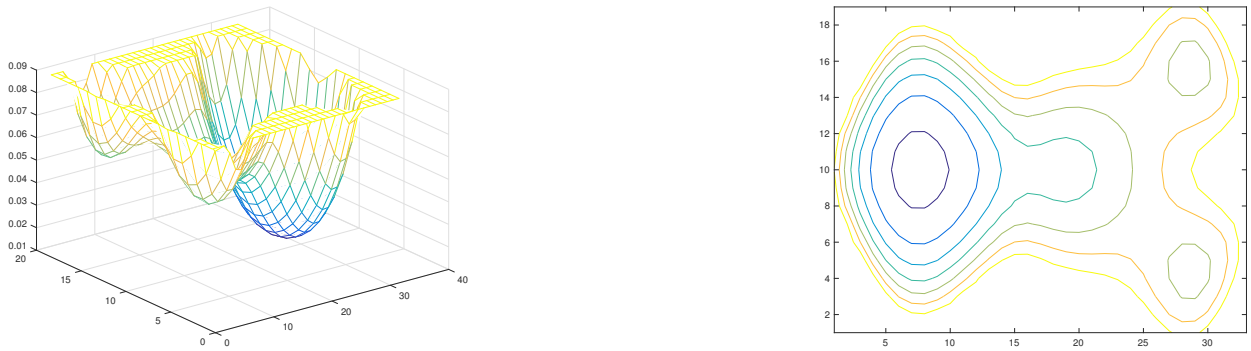

Figure 5.2: System 2 plots.

Example (3). Consider the $A \in \mathbb{R}^{5\times5}$ and $B \in \mathbb{R}^{5\times3}$ system with a "seed" of [1423]

$$A = \begin{bmatrix} 0.1387 & 0.0925 & 0.0926 & 0.4407 & 0.2900 \\ 0.3731 & 0.0193 & 0.9164 & 0.3564 & 0.3151 \\ 0.8792 & 0.8490 & 0.3542 & 0.4613 & 0.3271 \\ 0.4859 & 0.5388 & 0.1122 & 0.3825 & 0.9888 \\ 0.2255 & 0.7597 & 0.4938 & 0.8477 & 0.6068 \end{bmatrix}, B = \begin{bmatrix} 0.1387 & 0.0925 & 0.0926 \\ 0.3731 & 0.0193 & 0.9164 \\ 0.8792 & 0.8490 & 0.3542 \\ 0.4859 & 0.5388 & 0.1122 \\ 0.2255 & 0.7597 & 0.4938 \end{bmatrix}$$

Method	Results	
	Global Min	λ
Trisection	0.2249	-0.4671 + 0.0196i
Density search	0.2240	-0.4318
Two-phase	0.2271	-0.4016

Table 5.3: Results for system 3

In this example also, the least function value was attained by the Density search method. Below

is a 3-D and a contour plot of the system. Two local minima were encountered.

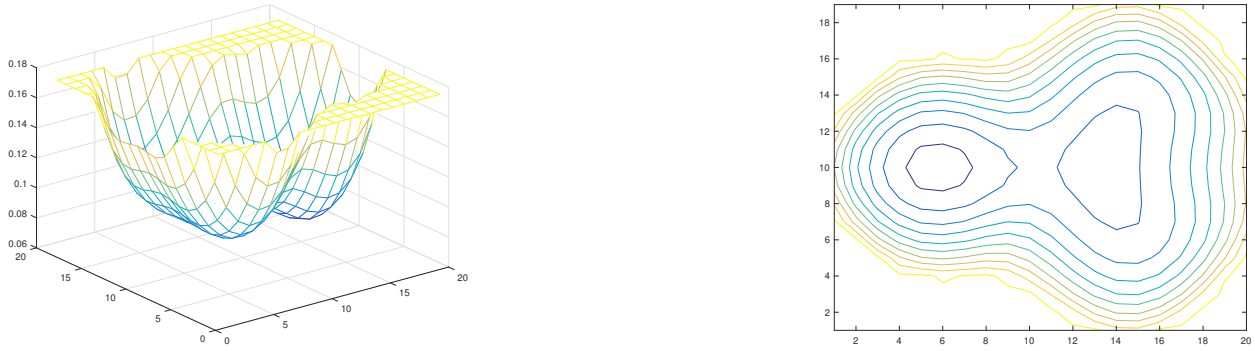

Figure 5.3: System 3 plots.

Example (4). Consider the $A \in \mathbb{R}^{6\times 6}$ and $B \in \mathbb{R}^{6\times 2}$ system with a "seed" of [1328]

$$A = \begin{bmatrix} 0.3602 & 0.8162 & 0.7422 & 0.8789 & 0.7530 & 0.9041 \\ 0.5107 & 0.8441 & 0.4996 & 0.6493 & 0.0981 & 0.2794 \\ 0.5049 & 0.0636 & 0.6720 & 0.6223 & 0.5953 & 0.7410 \\ 0.7333 & 0.1252 & 0.3899 & 0.6210 & 0.3009 & 0.9809 \\ 0.5986 & 0.7888 & 0.7413 & 0.5001 & 0.0285 & 0.6748 \\ 0.7821 & 0.6927 & 0.8684 & 0.1102 & 0.0641 & 0.5343 \end{bmatrix}, B = \begin{bmatrix} 0.3602 & 0.8162 \\ 0.5107 & 0.8441 \\ 0.5049 & 0.0636 \\ 0.7333 & 0.1252 \\ 0.5986 & 0.7888 \\ 0.7821 & 0.6927 \end{bmatrix}$$

Method	Results	
	Global Min	λ
Trisection	0.0632	-0.2299 + 0.0017i
Density search	0.0633	-0.2409
Two-phase	0.0633	-0.2277

Table 5.4: Results for system 4

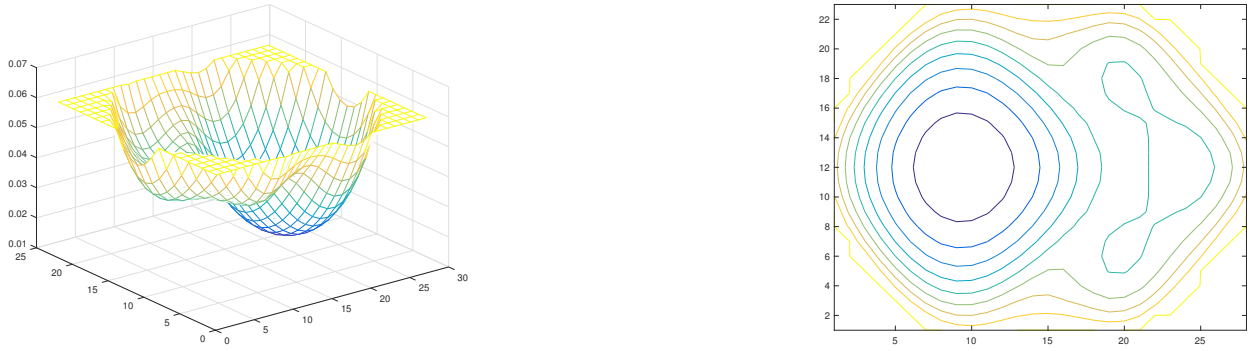

Figure 5.4: System 4 plots.

We also test the Two-phase algorithm on the numerical examples given in table (5.5)

size(n, m)(seed)	Trisection		Density Search		Two-phase	
	μ	λ	μ	λ	μ	λ
(5,1). (1423).	0.1113	-0.5812 + 0.0161i	0.1088	-0.5683	0.1138	-0.5087
(10,1). (1423).	0.0077	0.2976 + 0.0093i	0.0059	0.3004	0.0067	0.3012 + 0.0061i
(10,3). (1423).	0.0522	-0.0973 + 0.0445i	0.0483	-0.0650	0.0483	-0.0696 + 0.0067i
(15,1). (1423).	0.0098	-0.1453 + 0.0033i	0.0083	-0.1366	0.0085	-0.1382 + 0.0027i
(15,4). (1423).	0.0694	-0.1044 + 0.0199i	0.0672	-0.0848	0.0677	-0.0920 + 0.0105i
(17,1). (1423).	0.0119	-0.0909 + 0.0106i	0.0108	-0.0754	0.0109	-0.0737 + 0.0026i
(17,2). (1423).	0.0231	-0.0518 + 0.0104i	0.0215	-0.0379	0.0216	-0.0364 + 0.0018i
(19,1). (1423).	0.0001	0	0.0000	-0.0002	0.0000	-0.0002
(15,5). (1423).	0.0696	-0.1064 + 0.0200i	0.0672	-0.0858	0.0682	-0.0684 + 0.0060i
(21,1). (1423).	0.0120	0.5515 + 0.7080i	0.0105	0.5531 + 0.7005i	0.0326	0.6021 + 0.6915i
(25,1). (1423).	0.0083	-1.1041 + 0.0138i	0.0029	-1.0916	0.0030	-1.0907
(25,3). (1423).	0.0880	0.6450 + 0.0737i	0.0841	0.6990	0.0880	0.6729 + 0.1070i
(27,1). (1423).	0.0075	-0.3395 + 0.0057i	0.0070	-0.3352	0.0070	-0.3349

Table 5.5: Algorithm comparison for the various systems

General observation of the Two-phase algorithm is that, it converges faster when there is a good starting value for the tunneling phase. When a good starting point is used, then the tunneling phase takes lesser time to converge to a point $\hat{\lambda}$, for the minimization phase to begin. This in turn reduces the number of singular value decomposition evaluation.

The table below shows the average number of times the minimum singular value and its respective singular vectors were evaluated for some chosen numerical problems.

size(n,m)	Density Search		Two-phase	
	Aveg. σ_n Eval	Aveg. u_n,v_n Eval	Aveg. σ_n Eval.	Aveg.u_n,v_n Eval.
(5,1).	507	8	10536	20103
(10,1).	978	14	17656	25374
(15,1).	4342	53	24533	32746
(20,1).	5423	84	34316	44316
(25,1).	10232	94	41325	59325
(30,1).	6243	337	59649	73164

Table 5.6: Average singular value decomposition evaluation counts

5.1.1 Average Algorithm timings

In this section, we evaluate the execution timings for the three algorithms. We compare the CPU time taken by each algorithm to execute a particular problem depending on the size of the systems.

size(n,m)	t_{cpu} (Trisection)	t_{cpu} (Density Search)	t_{cpu} (Two-phase Time)
(5,1)	1.9799	0.6499	2.7782
(10,1)	3.8513	0.7435	9.3893
(15,1)	9.0586	0.9370	18.9598
(20,1)	15.6234	1.3078	29.2957
(25,1)	30.5623	3.6767	60.0142
(30,1)	56.453	4.3342	76.4925
(35,1)	65.452	5.0324	90.3945

Table 5.7: Average Time with respect to order of the System

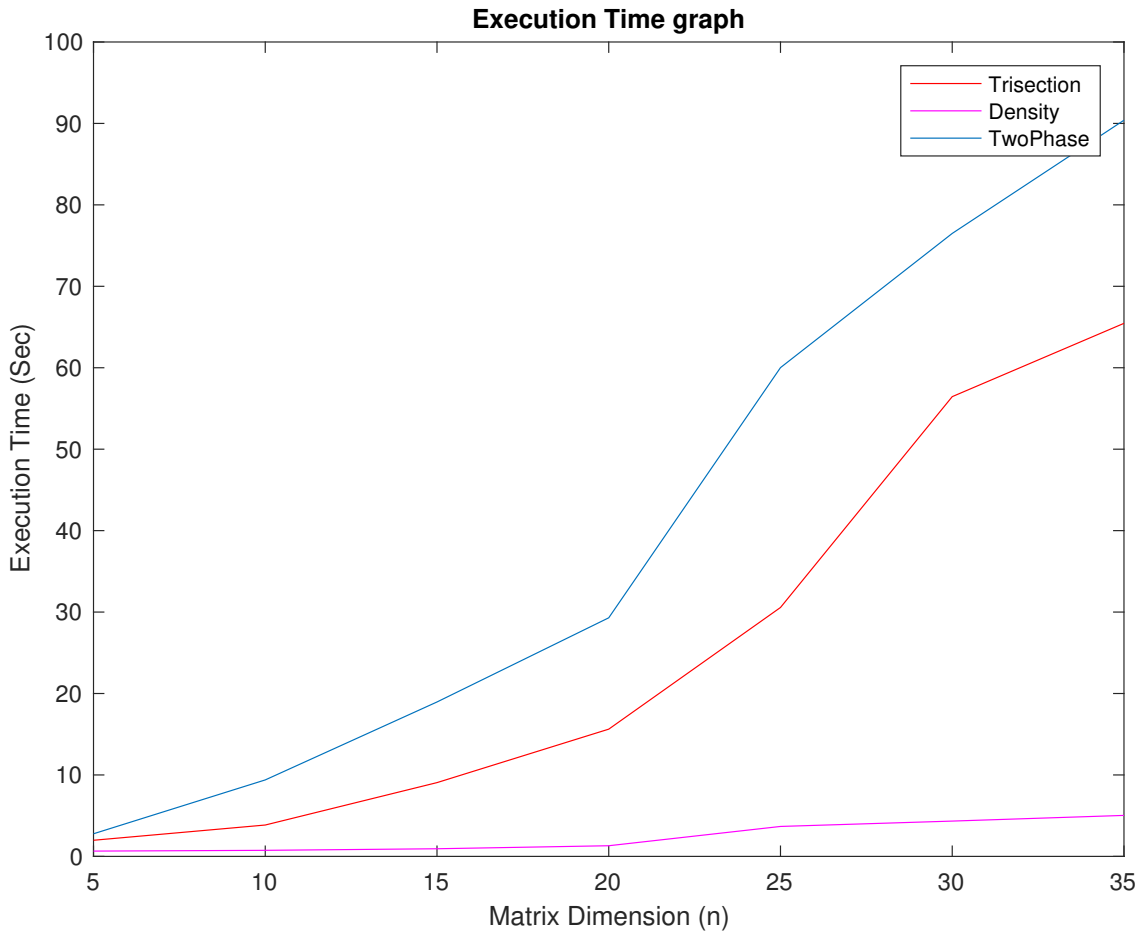

Figure 5.5: Graph of execution time vs n.

In this category, the Two-phase algorithm average time is not satisfactory as compared to the rest. The fastest algorithm seems to be the Density search method. Trisection method is not performing badly also at least when compared to Two-phase. Much of the computational time in the Two-phase algorithm is spent in choosing appropriate pole strength in the Newton's phase which computes the zero point of the tunneling function. Also finding an initial value which brings Newton's method to convergence is another problem which we encountered during the execution processes.

5.1.2 Performance of Tunneling algorithm used for Real search

Here, it is notable to mention the performance of the Tunneling algorithm implemented on the case for $\lambda \in \mathbb{R}$. Implementing this algorithm was rather simple since the derivative of the function (1.5) can easily be computed using equation (3.9) which in turn makes it easier to use optimization algorithms in both the minimization phase and the tunneling phase. We used the gradient descent method in the minimization phase and the Newton's method in the tunneling phase. Numerical examples confirm that, the algorithm reaches the same or close to the global minimum when we compared with Density search algorithm for $\lambda \in \mathbb{R}$. Below we present some numerical examples for matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times 1}$. These computed results represent the "scaled" system.

size(n, m) (seed)	Density Search		Two - Phase	
	Global Min	λ	Global Min	λ
(5,1). (1430).	0.0033	0.0100	0.0033	0.0100
(10,1). (1430)	0.0023	0.0456	0.0041	0.0653
(15,1). (1430)	0.0039	0.1139	0.00483	0.1249
(20,1). (1430)	2.2498e-04	0.0106	7.7232e-04	-0.0392
(25,1).(1430)	2.1305e-05	0.0932	4.6209e-04	0.0964
(30,1). (1430)	7.0468e-05	-0.0252	6.3876e-05	-0.0076

Table 5.8: Results obtained when $\lambda \in [-\eta, \eta]$

In [16] it is shown that there can be a complex λ that can give the smallest 2-norm real perturbation so that the perturbed system is uncontrollable. In this method, unlike ours, perturbations of rank higher than one are considered.

6 Conclusions

The purpose of this thesis was to study an algorithm that uses both Density search algorithm [25] and Tunneling method [21] to determine the distance of a linear system from the nearest uncontrollable system. And as it turned out the problem of computing this distance was equivalent to the global optimization problem

$$\mu = \min_{\lambda \in \mathbb{C}} \sigma \left(B, A - \lambda I \right)$$

We also considered the global optimization problem

$$\mu = \min_{\lambda \in \mathbb{R}} \sigma \left(B, A - \lambda I \right)$$

We call this algorithm Two-phase algorithm. Results were compared with Density search algorithm [25] and Trisection algorithm by [23] which uses the test scheme by [14]. In this algorithm, no new theories were formulated or discovered, but the idea was to combine these two existing algorithms to implement a new algorithm that compute the global minimum. Interestingly from observations based on numerical results, this algorithm gives a good estimation to the global minimum when compared with other algorithms. However this method turned out to be an expensive method based on the number of singular value decomposition evaluations involved. For the purpose of research, the algorithm achieves its goal by giving a good estimation of the global minimum.

7 Basic Notations

μ : Distance to uncontrolllability of the system A and B.

μ_r : Distance to uncontrolllability of the system A and B in $\mathbb{R}$.

μ_c : Distance to uncontrolllability of the system A and B in $\mathbb{C}$.

$\sigma(\lambda)$: Minimum singular value of the system A and B.

$\mathbb{R}$: Field of Real numbers.

$\mathbb{C}$: Field of Complex numbers.

$\mathbb{R}^n$: Vectors of real number of size n. F : Integers.

$\mathbb{C}^n$: vectors of complex numbers of size of n.

$\mathbb{R}^{n\times m}$: real $n \times m$ matrices.

$\mathbb{C}^{n\times m}$: complex $n \times m$ matrices.

$||A||$: 2-norm of the matrix A.

$||A||_F$: Frobenius norm of the matrix A.

$\lambda_{min}(A)$: smallest eigenvalue of the Hermitian matrix A.

$\lambda_{max}(A)$: largest eigenvalue of the Hermitian matrix A.

$\sigma(A)$: Set of singular values of the matrix A.

$\sigma_n(A)$: minimum singular value of the $n \times m$ matrix A.

u_n : Left singular vector corresponding to the minimum singular value σ_n.

v_n : Right singular vector corresponding to the minimum singular value σ_n.

α_{nn} : N-th element of matrix A.

Re c : Real part of the complex number c.

Im c : Imaginary part of the complex number c.

Bibliography

[1] D. Boley. "Computing rank-deficiency of rectangular matrix pencils". *Systems & Control Letters*, 9(3):207–214, sep 1987.

[2] D. Boley and W.-S. Lu. Measuring how far a controllable system is from an uncontrollable one. *IEEE Transactions on Automatic Control*, 31(3):249–251, 1986.

[3] D. L. Boley. Computing rank-deficiency of rectangular matrix pencils. In *1986 25th IEEE Conference on Decision and Control*, pages 554–557, Dec 1986.

[4] S. J. Boyce, L. Zietsman, A. H. Norton, J. T. Borggaard, and M. V. Day. "The Distance to Uncontrollability via Linear Matrix Inequalities". 2010.

[5] J. W. Brown, R. V. Churchill, and M. Lapidus. *"Complex variables and applications"*, volume 7. McGraw-Hill New York, 1996.

[6] J. W. Brown et al. *Complex variables and applications.* Boston: McGraw-Hill Higher Education, 2009.

[7] J. V. Burke, A. S. Lewis, and M. L. Overton. "Pseudospectral components and the distance to uncontrollability ". *Society*, 26:350–361, 2004.

[8] R. Byers. Detecting nearly uncontrollable pairs. In *Numerical methods proceedings of the international symposium MTNS-89*, volume 3, pages 447–457, 1990.

[9] L. Castellanos and S. Gomez. A new implementation of the tunneling methods for bound constrained global optimization. *Reporte de Investigacin IIMAS*, 10(59):1–18, 2000.

[10] R. Eising. "Between controllable and uncontrollable". *Systems & control letters*, 4(5):263–264, 1984.

[11] R. M. Freund. "The Steepest Descent Algorithm for Unconstrained Optimization and a Bisection Line-search Method". *ReCALL*, pages 23–25, 2004.

[12] M. Gao and M. Neumann. A global minimum search algorithm for estimating the distance to uncontrollability. *Linear algebra and its applications*, 188:305–350, 1993.

[13] G. H. Golub and C. F. Van Loan. "Matrix computations. Johns Hopkins studies in the mathematical sciences", 1996.

[14] M. Gu. New methods for estimating the distance to uncontrollability. *SIAM Journal on Matrix Analysis and Applications*, 21(3):989–1003, 2000.

[15] M. T. Heath. *Scientific computing*. McGraw-Hill New York, 2002.

[16] G. Hu and E. J. Davison. A real radius measure for controllability. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 4, pages 3144–3148 vol.4, 2001.

[17] C. Kenney and A. J. Laub. "Controllability and stability radii for companion form systems". *Mathematics of Control, Signals and Systems*, 1(3):239–256, 1988.

[18] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.

[19] S. Lam and E. J. Davison. Computation of the real controllability radius and minimum-norm perturbations of higher-order, descriptor, and time-delay lti systems. *IEEE Transactions on Automatic Control*, 59(8):2189–2195, Aug 2014.

[20] L.Elsner and C.He. "An algorithm for computing the distance to uncontrollability". 17:453–464, 1991.

[21] A. V. Levy and A. Montalvo. "The tunneling algorithm for the global minimization of functions". *SIAM Journal on Scientific and Statistical Computing*, 6(1):15–29, 1985.

[22] D. Luenberger. "Introduction to dynamic systems: theory, models, and applications". 1979.

[23] E. Mengi. *Measures for robust stability and controllability.* PhD thesis, Courant Institute of Mathematical Sciences New York, 2006.

[24] G. Miminis. "Numerical algorithms for controllability and eigenvalue allocation", master's thesis. *McGill University, School of Computer Science, Montreal, PQ Canada H3A 2B5*, 1981.

[25] G. Miminis. Course notes in numerical control, May 2015.

[26] W. K. Nicholson. *Elementary linear algebra, with applications.* PWS-Kent Publishing Company, 1990.

[27] C. C. Paige. "Properties of numerical algorithms related to computing controllability". *Automatic Control, IEEE Transactions on*, 26(1):130–138, 1981.

[28] A. Rinnooy Kan, C. Boender, and G. T. Timmer. A stochastic approach to global optimization. 1984.

[29] H. Schwetlick, U. Schnabel, and P.-R. I. der DFG-Forschergruppe. *Iterative computation of the smallest singular value and the corresponding singular vectors of a matrix.* Citeseer, 1997.

[30] G. W. Stewart. *Introduction to matrix computations.* Computer science and applied mathematics. Academic Press, 1973.

[31] J. Stewart. *Calculus.* Cengage Learning, 2015.

[32] M. Wicks and R. A. Decarlo. "Computing the Distance to an Uncontrollable System". 36(1):39–49, 1991.