A fast algorithm to compute the controllability, decentralized fixed-mode, and minimum-phase radius of LTI systems

Simon Lam and Edward J. Davison

Abstract—In this paper, an efficient algorithm is presented for solving the 2-D optimization problem which is associated with computing the robustness measures of a number of linear time-invariant (LTI) system properties such as the controllability radius, the decentralized fixed-mode radius, and the minimum-phase radius. Unlike methods such as gradient search methods, the proposed algorithm works well independent of the initial trial point, and obtains the minimum with a high level of confidence that it is indeed global. A numerical example is included.

I. INTRODUCTION

Consider the following LTI multivariable system

\[
\dot{x} = Ax + Bu \\
y = Cx + Du
\]

where \(x \in \mathbb{R}^n\), \(u \in \mathbb{R}^m\), and \(y \in \mathbb{R}^r\) are respectively the state, input, and output vectors, and \(A, B, C,\) and \(D\) are constant matrices with the appropriate dimensions for \(n \geq 1\), \(m \geq 1, r \geq 1,\) and \(\max(r, m) \leq n\). It is well known that there exists a LTI controller that can assign the eigenvalues of the closed-loop system to any arbitrary spectrum if and only if the system is controllable and observable. However, when a controllable system is subject to parametric perturbations (i.e. \(A \rightarrow A + \Delta_A \) and \(B \rightarrow B + \Delta_B\)), the system may be very “close” to becoming uncontrollable. Hence, a continuous controllability measure is more informative than the traditional ‘yes/no’ controllability metric, which simply determines whether a system is controllable or not. The same can be said about other system properties such as observability, stability, minimum-phase, etc.

In the current literature, various continuous measures have been proposed to measure the robustness of various system properties with respect to parametric perturbations. In particular, the controllability radius introduced in [1], [2] measures how close a controllable system is to being uncontrollable, the stability radius in [3] measures how close a stable system is to an unstable one, and the decentralized fixed-mode (DFM) radius ([4], [5]) measures how close a system with no DFMs is to having one. More recently, such robustness measures have been extended to characterize the robustness of a system’s transmission zero properties. For instance, the minimum-phase radius in [6] measures how close a minimum-phase system is to becoming non-minimum phase.

Computing the stability radius mentioned above ([3]) requires solving only a 1-D optimization problem, and a fast algorithm to solve this is proposed in [7]. On the other hand, computing the other radii mentioned above (i.e. except the stability radius) requires solving a 2-D optimization problem.

As of now, the 2-D problem is solved using nonlinear search methods such as gradient search methods. A disadvantage with using such nonlinear techniques is that the minimum obtained may be a local minimum. Hence selecting the initial trial points are critical, and a large number of initial points are generally tested in order to achieve a certain level of confidence that the obtained minimum is global. In this paper, we propose a more efficient algorithm for solving the 2-D optimization problem. The proposed algorithm has some similar features as the one in [7] (in particular, the use of a so-called “minimizing set”), and has the advantage that it is not dependent on the initial test values, and obtains the global minimum with a high degree of confidence.

The paper is organized as follows. In Section II, the controllability radius is reviewed and the 2-D optimization problem to be solved is described. Then in Section III, the few useful tools are developed, followed by a thorough description of the algorithm in Section IV. Finally, Section V gives a numerical example to demonstrate the effectiveness of the algorithm.

II. NOTATION AND BACKGROUND

In this paper, the field of real and complex numbers are denoted by \(\mathbb{R}\) and \(\mathbb{C}\) respectively. \(\mathbb{C}_+\) and \(\mathbb{C}_U\) denotes the closed right half and the closed upper half of the complex plane respectively, and \(\mathbb{R}_+\) denotes the nonnegative real line \([0, +\infty)\). The \(i\)-th singular value of a matrix \(M \in \mathbb{C}^{p \times m}\) is denoted by \(\sigma_i(M)\), where \(\sigma_1(M) \geq \sigma_2(M) \geq \cdots \geq \|M\|\) denotes the spectral norm of a matrix \(M\) and is equal to \(\sigma_1(M)\). Also, \(\overline{M}, M^T, M^H,\) and \(M^+\) denote respectively the complex conjugate, transpose, complex conjugate transpose, and Moore-Penrose pseudoinverse of \(M\). The real and imaginary components of the matrix \(M\) are given by \(\Re M\) and \(\Im M\) respectively. Finally, the set of eigenvalues of a square matrix \(A \in \mathbb{C}^{n \times n}\) is denoted by \(\lambda(A)\).

A. Controllability radius

The following definition is made.

Definition 2.1: Given a LTI system (1), the controllability radius, \(r_c\), is defined to be:

\[
r_c(A, B) = \inf\{\|\Delta_A, \Delta_B\| | \Delta_A \in \mathbb{F}^{n \times n}, \Delta_B \in \mathbb{F}^{n \times m}, (A + \Delta_A, B + \Delta_B) \text{ is uncontrollable}\}\]

\[\text{(2)}\]
where \( F \in \{ \mathbb{C}, \mathbb{R} \} \). Similarly, the stabilizability radius, \( r_{\mathbb{F}}^{c+} \), is defined as:

\[
r_{\mathbb{F}}^{c+}(A, B) = \inf \{ \| \Delta A, \Delta B \| : \Delta A \in \mathbb{F}^{n \times n}, \Delta B \in \mathbb{F}^{n \times m}, (A + \Delta A, B + \Delta B) \text{ is unistablizible} \}
\]  

(3)

In [1] and [2], the complex and real controllability radii are respectively shown to be given by:

\[
r_{c}^{c}(A, B) = \min_{s \in \mathbb{C}} \sigma_{n}(\begin{bmatrix} A - sI_{n} & B \end{bmatrix})
\]  

(4)

and

\[
r_{c}^{r}(A, B) = \min_{s \in \mathbb{C}} \tau_{n}(\begin{bmatrix} A - sI_{n} & B \end{bmatrix})
\]  

(5)

where for \( W \in \mathbb{C}^{p \times q} \),\(^{2}\)

\[
\tau_{n}(W) := \sup_{\gamma \in (0, 1]} \sigma_{2n-1}(\begin{bmatrix} \text{Re} W & -\gamma \text{Im} W \\ -\gamma^{-1} \text{Im} W & \text{Re} W \end{bmatrix})
\]  

(6)

For notational convenience in the paper, denote

\[
P(\gamma, W) := \begin{bmatrix} \text{Re} W & -\gamma \text{Im} W \\ -\gamma^{-1} \text{Im} W & \text{Re} W \end{bmatrix}
\]  

(7)

Similarly, the complex and real stabilizability radii are respectively shown to be:

\[
r_{c}^{c+}(A, B) = \min_{s \in \mathbb{C}^{+}} \sigma_{n}(\begin{bmatrix} A - sI_{n} & B \end{bmatrix})
\]  

(8)

and

\[
r_{c}^{r+}(A, B) = \min_{s \in \mathbb{C}^{+}} \tau_{n}(\begin{bmatrix} A - sI_{n} & B \end{bmatrix})
\]  

(9)

\section*{B. 2-D Optimization Problem}

It can be seen from (4)–(9) that a 2-D optimization problem in the complex plane is required to be solved when computing the complex and real controllability/stabilizability radii. A similar 2-D problem is to be solved when computing the DFM radius ([4], [5]), and the minimum-phase radius ([6]). In this paper, a fast algorithm is proposed to solve the two standard general problems associated with such 2-D optimization problems; i.e. for given constant real matrices \((A, B, C, D)\) and \(i \in \{1, \ldots, n + \min(r, m)\} \):

Problem 1: Find

\[
r_{c}^{i}(C, A, B, D) = \min_{s \in \mathbb{C}} \sigma_{i}(\begin{bmatrix} A - sI_{n} & B \\ C & D \end{bmatrix})
\]  

(10)

and

Problem 2: Find

\[
r_{r}^{i}(C, A, B, D) = \min_{s \in \mathbb{C}} \tau_{i}(\begin{bmatrix} A - sI_{n} & B \\ C & D \end{bmatrix})
\]  

(11)

For the sake of simplicity, denote \( G(s) := \begin{bmatrix} A - sI_{n} & B \\ C & D \end{bmatrix} \).

To avoid the trivial case where the radius is zero for all \( s \in \mathbb{C} \), we will assume that the two problems (10) and (11) satisfy the following assumption:

\[
\text{rank}(\begin{bmatrix} A & B \\ C & D \end{bmatrix}) \geq i
\]  

(12)

\footnote{To distinguish which field is consider, \( r_{\mathbb{F}}^{c} \) (i.e. \( F = \mathbb{C} \)) is called the complex controllability radius and \( r_{\mathbb{F}}^{r} \) is called the real controllability radius.}

\footnote{\( \tau_{n}(W) \) is referred to as the \( n \)-th real perturbation value of \( W \) (see [8]).}

\section*{III. Preliminary Results}

Before presenting an efficient algorithm for solving (10) and (11), some preliminary tools are needed; in particular Theorem 3.1 & 3.2 given below.

\textbf{Definition 3.1 (The \( \mathcal{H} \) matrix):} Given \( x \geq 0, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times n}, C \in \mathbb{R}^{n \times n}, D \in \mathbb{R}^{n \times m}, \) and nonsingular matrices \( \Delta \in \mathbb{C}^{n \times n}, E \in \mathbb{C}^{(n+r) \times (n+r)}, \) and \( F \in \mathbb{C}^{(n+m) \times (n+m)}, \) define:

\[
\mathcal{H}(x, A, B, C, D, \Delta, E, F) := \Delta^{-1}(\tilde{A} - \tilde{B}\tilde{D}^{-1}\tilde{C})
\]  

(13)

where \( \tilde{A} = \begin{bmatrix} A & 0 \\ 0 & A^{T} \end{bmatrix} + T_{11}, \tilde{B} = \begin{bmatrix} B & 0 \\ 0 & C^{T} \end{bmatrix} + T_{12}, \)

\( \tilde{C} = \begin{bmatrix} C & 0 \\ 0 & B^{T} \end{bmatrix} + T_{21}, \tilde{D} = \begin{bmatrix} D & 0 \\ 0 & D^{T} \end{bmatrix} + T_{22}, \) and

\( \tilde{\Delta} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta^{H} \end{bmatrix} \), and where \( T := \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \) satisfies

\[
\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = P_{t} \begin{bmatrix} -x(E^{H})^{-1} & 0 \\ 0 & -x(FF^{H})^{-1} \end{bmatrix} P_{t}
\]  

(14)

for the permutation matrices \( P_{t} = \begin{bmatrix} I_{n} & 0 & 0 \\ 0 & I_{n} & 0 \\ 0 & 0 & I_{m} \end{bmatrix} \) and \( P_{r} = \begin{bmatrix} 0 & I_{n} & 0 \\ 0 & 0 & I_{r} \\ 0 & 0 & 0 \end{bmatrix} \).

The following preliminary result is obtained.

\textbf{Theorem 3.1:} Let \( M := E \begin{bmatrix} A - s\Delta & B \\ C & D \end{bmatrix} \), where \( s \in \mathbb{R} \), and \( E \in \mathbb{C}^{(n+r) \times (n+r)}, F \in \mathbb{C}^{(n+m) \times (n+m)}, \) and \( \Delta \in \mathbb{C}^{n \times n} \) are all nonsingular. Then, for given \( x \in \mathbb{R}_{+}, \)

\[
x \in \sigma(M) \Leftrightarrow s \in \lambda(\mathcal{H}(x, A, B, C, D, \Delta, E, F))
\]  

(15)

Proof:

\[
x \in \sigma(M) \Leftrightarrow x \in \lambda(\begin{bmatrix} 0 & M \\ M^{H} & 0 \end{bmatrix})
\]  

\[\Leftrightarrow \text{det}(\begin{bmatrix} -xI_{n+r}^{H} & M \\ M^{H} & -xI_{n+m} \end{bmatrix}) = 0 \]

\[\Leftrightarrow \text{det}(P_{t}(\begin{bmatrix} -xI_{n+r}^{H} & M \\ M^{H} & -xI_{n+m} \end{bmatrix}P_{r})) = 0 \]

\[\Leftrightarrow \text{det}(\begin{bmatrix} A - s\Delta & 0 \\ 0 & A^{T} - s\Delta^{H} \end{bmatrix}) = 0 \]

\[\Leftrightarrow \text{det}(\begin{bmatrix} A - s\Delta \\ C \end{bmatrix}) = 0 \]

\[\Leftrightarrow s \in \lambda(\Delta^{-1}(\tilde{A} - \tilde{B}\tilde{D}^{-1}\tilde{C})) \]


Remark 3.1: If $\tilde{D}$ in (13) is singular, then by (15), (14) can be modified such that $s$ in (14) is a generalized eigenvalue of the matrix pair $\left(\begin{array}{cc} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{array}\right)$ instead; i.e. $s$ is in the set of $\lambda \in \mathbb{C}$ such that $\left[\begin{array}{cc} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{array}\right] x = \lambda \left[\begin{array}{c} \tilde{\Delta} \\ 0 \\ 0 \end{array}\right] x$ for some non-zero eigenvector $x \in \mathbb{C}^{2n+m+r}$.

The following result is also obtained.

Theorem 3.2: Let $\gamma \in (0,1]$, $\theta \in [0,2\pi]$ and real $x \geq 0$ be given. Then, for all $s \in \mathbb{R}_+$,

$$x \in \sigma(P(\gamma, G(se^{i\theta}))) \iff s \in \lambda(\mathcal{H}(x, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{T}_{\gamma,n+1}^{-1}, P_{nr}, P_{nm}T_{\gamma,n+m}))$$

where $\tilde{A} = \left[\begin{array}{cc} A & 0 \\ 0 & A \end{array}\right]$, $\tilde{B} = \left[\begin{array}{cc} B & 0 \\ 0 & B \end{array}\right]$, $\tilde{C} = \left[\begin{array}{cc} C & 0 \\ 0 & C \end{array}\right]$, $\tilde{D} = \left[\begin{array}{cc} D & 0 \\ 0 & D \end{array}\right]$, $\tilde{\Delta} = \left[\begin{array}{cc} e^{i\theta}I_n & 0 \\ 0 & e^{-i\theta}I_n \end{array}\right]$, $P_{nr} = \left[\begin{array}{cc} I_n & 0 \\ 0 & I_r \end{array}\right]$, $P_{nm} = \left[\begin{array}{cc} I_n & 0 \\ 0 & I_m \end{array}\right]$, and

$$T_{\gamma,n} := \frac{1}{2\pi} \left[\begin{array}{cc} \sqrt{1+\gamma^2}I_n & i\gamma \sqrt{1+\gamma^2}I_n \\ i\gamma \sqrt{1+\gamma^2}I_n & -\gamma \sqrt{1+\gamma^2}I_n \end{array}\right].$$

Proof: It can easily be verified (e.g. see [8]) that

\[
T_{\gamma,n+1}P(\gamma, G(se^{i\theta})) T_{\gamma,n+1}^{-1} = \begin{bmatrix} A - se^{i\theta}I_n & 0 & B \\ 0 & A - se^{-i\theta}I_n & 0 \\ C & 0 & D \end{bmatrix}
\]

Therefore,

\[
P(\gamma, G(se^{i\theta})) = T_{\gamma,n+1} P_{nr} \begin{bmatrix} A - s\tilde{\Delta} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} P_{nm} T_{\gamma,n+1}
\]

and the proof follows immediately from Theorem 3.1.

IV. ALGORITHM

An efficient iterative algorithm for solving (10) and (11) for given constant real matrices $(A,B,C,D)$ and $i \in \{1,\ldots,n+\min(m,r)\}$ is now presented in this section. The following development focuses mainly on solving (11), but it will be shown later that with a slight modification, the algorithm also applies to solving (10).

Firstly, let the global minimum of (11) be denoted by:

$$r^* = \min_{s \in \mathbb{C}} \tau_i(G(s)) = \min_{s \in \mathbb{C}} \sup_{\gamma \in [0,1]} \sigma_{2i-1}(P(\gamma^*, G(s)))$$

and is achieved at $s^* \in \mathbb{C}$ and $\gamma^* \in (0,1]$, i.e. $r^* = \sigma_{2i-1}(P(\gamma^*, G(s^*)))$. Now suppose that at the $k$-th iteration, for $k = 1,2,\ldots$, we are given $r_{k-1}$, $\gamma_{k-1}$, and $s_{k-1}$ (i.e. obtained from the previous iteration), which are approximations of $r^*$, $\gamma^*$, and $s^*$ respectively, where $r_{k-1} = \sigma_{2i-1}(P(\gamma_{k-1}, G(s_{k-1})))$. Furthermore, suppose we are also given the so-called “minimizing set”, $S_{k-1}$, which is a (closed) set in the complex plane that contains the global minimizer $s^*$. The basic idea of the algorithm is then to use the information of $r_{k-1}$, $\gamma_{k-1}$, and $s_{k-1}$, together with Theorem 3.2, to reduce the size of $S_{k-1}$; i.e. to reduce the region containing the global minimizer $s^*$. Denoting the reduced set by $S_k (\subseteq S_{k-1})$, we then search within $S_k$ for a new point $s_k \in S_k$ such that

$$\tau_i(G(s_k)) < r_{k-1}$$

and assign $r_k = \tau_i(G(s_k))$. Also, assign $\gamma_k$ to be the value of $\gamma$ that achieves $r_k$; i.e. $r_k = \sigma_{2i-1}(P(\gamma_k, G(s_k)))$. Note that since $r_k < r_{k-1}$, then $r_k$ is a new and better approximation of the global minimum $r^*$. The procedure is then repeated until either the size of $S_k$ is smaller than an user-specified tolerance, or until $r_k$ becomes “very close” to zero (more on this later).

A. The minimizing set, $S_k$

In the algorithm, the “minimizing set” is defined to be the union of particular sectors of interest in the closed upper half of the complex plane, and can be described by $S_k \subseteq [0,\pi]$. The reason why the “minimizing set” is constrained to the closed upper half of the complex plane is due to the following (e.g. see [6]):

$$\tau_i\left(\begin{bmatrix} A - sI & B \\ C & D \end{bmatrix}\right) = \tau_i\left(\begin{bmatrix} A - sI & B \\ C & D \end{bmatrix}\right)$$

In other words, the radii in (10) and (11) are both symmetrical with respect to the real axis; hence one only needs to search within either the closed upper or lower half of the complex plane for the global minimum.

It should be pointed out that $S_k$ is a subset of $[0,\pi]$, and that the underlying “minimizing set” actually consists of the sectors in the complex plane described by $S_k$; i.e. the “minimizing set” is $\{s \in \mathbb{C} | s \in S_k\}$, where $S_k$ denotes the angle of $s \in \mathbb{C}$ For convenience, however, we will sometimes refer to the “minimizing set” as $S_k$.

To reduce the size of $S_k$ based on given $r_{k-1}$, $\gamma_{k-1}$, and $s_{k-1}$, where $r_{k-1} = \sigma_{2i-1}(P(\gamma_{k-1}, G(s_{k-1})))$, we first construct the set $R_k$:

$$R_k = \{s \in \mathbb{C}_{i\ell} | \sigma_{2i-1}(P(\gamma_{k-1}, G(s))) < r_{k-1}\}$$

The significance of $R_k$ is that for all points not in $R_k$, the radius can never be smaller than $r_{k-1}$. This is because for all $s \notin R_k$,

$$\tau_i(G(s)) = \sup_{\gamma \in [0,1]} \sigma_{2i-1}(P(\gamma, G(s))) \geq \sigma_{2i-1}(P(\gamma_{k-1}, G(s))) \geq r_{k-1}$$

Hence, one only needs to search within $R_k$ to find points in the complex plane that achieve a radius smaller than $r_{k-1}$.

3Note that $R_k$ can also be a (tighter) “minimizing set”. For ease of implementation, however, we chose the “minimizing set” to be the sectors containing $R_k$ (i.e. as described by $S_k$) instead.
To obtain $R_k$, we first fix a particular $\theta \in [0, \pi]$, and then find the set
\[ R_k^0 = \{ w \in \mathbb{R}_+ | \sigma_{2i-1}(P(\gamma_{k-1}, G(e^{i\theta}))) < r_{k-1} \} \]  \hspace{1cm} (16)

Therefore,
\[ R_k = \bigcup_{\theta \in [0, \pi]} R_k^\theta \]  \hspace{1cm} (17)

Note that $R_k^\theta$ for a particular $\theta \in [0, \pi]$ can be obtained by applying Theorem 3.2. In particular, by Theorem 3.2, the values along the ray, $R_\theta := \{ se^{i\theta} \in \mathbb{C} | s \in \mathbb{R}_+ \}$, that achieve a radius equal to $r_{k-1}$ are among the real nonnegative eigenvalues of $\mathcal{H}(r_{k-1}, A, B, \hat{C}, \hat{D}, \hat{\Delta}, T_{\gamma_{k-1}, n+r} P_{nr}, P_{nm} T_{\gamma_{k-1}, n+m})$. Hence, step (2-f) for solving problem (10) can very easily be accomplished by selecting any point in $R_k$.

D. Algorithm outline

The algorithm can be summarized as follows: Algorithm 4.1:

Input: $(A, B, C, D)$ and $i$, where (12) is satisfied

1) Initialization:
   - Set $S_0 = [0, \pi]$.
   - Choose an arbitrary $s_0 \in \mathcal{U}$, and compute $r_0$ and $\gamma_0$, where $r_0 = \sigma_{2i-1}(P(\gamma, G(s_0)))$.

2) Iteration $k (=1, 2, \ldots )$
   a) Given $r_{k-1}$, $\gamma_{k-1}$, $s_{k-1}$, and $S_{k-1}$.
   b) Reset $\Theta_k := \emptyset$.
   c) For $\theta \in S_{k-1}$ (\theta can be discretized steps of $S_{k-1}$)
      - Compute $R_k^\theta$ by finding all real nonnegative eigenvalues
        \[ s \in \lambda \left( \mathcal{H}(r_{k-1}, A, B, \hat{C}, \hat{D}, \hat{\Delta}, T_{\gamma_{k-1}, n+r} P_{nr}, P_{nm} T_{\gamma_{k-1}, n+m}) \right) \]
        such that $\sigma_{2i-1}(P(\gamma_{k-1}, G(s))) = r_{k-1}$.
      - If $R_k^\theta \neq \emptyset$, then update $\Theta_k := \Theta_k \cup \theta$.
   d) Update $S_k := S_{k-1} \cap \Theta_k$.
   e) Quit if the length of the largest interval of $S_k < TOL_{S_k}$ and $size(R_k) < TOL_{R_k}$.
   f) Find one point $s_k \in R_k^\theta$ such that $r_k < r_{k-1}$, where $\theta \in S_k$, and $r_k = \tau_i(G(s_k))$.

3) Quit if $r_k < TOL_{R_k}$.
4) Update $k := k + 1$ and goto step 2.

E. Algorithm for solving problem (10)

To solve problem (10), only two slight modifications of Algorithm 4.1 are needed, which actually results in a simpler algorithm. Firstly, instead of computing the eigenvalues of $\mathcal{H}(r_{k-1}, A, B, \hat{C}, \hat{D}, \hat{\Delta}, T_{\gamma_{k-1}, n+r} P_{nr}, P_{nm} T_{\gamma_{k-1}, n+m})$ in step (2-c) to obtain $R_k^\theta$, one is only required to compute the eigenvalues of $\mathcal{H}(r_{k-1}, A, B, C, D, e^{i\theta} I_{nr}, I_{nm} + I_{nr} n)$ (the proof is trivial and follows directly from Theorem 3.1). Furthermore, when solving problem (10), all the values within $R_k$ actually achieve a smaller radius than $r_{k-1}$ (and all the values outside $R_k$ do not). This is not true when solving problem (11), where $R_k$ only achieves a bound. Hence, step (2-f) for solving problem (10) can very easily be accomplished by selecting any point in $R_k$.
F. Algorithm for computing the real stabilizability radius

To compute the real stabilizability radius, the real unstable DFM radius, and the real minimum-phase radius, where the search is for a minimum in the closed right half of the complex plane – i.e. by solving the following problem

\[
\hat{r}_i^\tau (A, B, C, D) = \min_{s \in C^+} \tau_i \left( \begin{bmatrix} A - sI & B \\ C & D \end{bmatrix} \right) \tag{20}
\]

for given \((A, B, C, D)\) and \(i \in \{1, \ldots, n + \min(r, m)\}\) – only one very simply modification of Algorithm 4.1 is needed. In particular, one only needs to change the initialization step (1) of Algorithm 4.1 from \(S_0 = [0, \pi] \) to \(S_0 = [0, \frac{\pi}{2}] \) (the proof is trivial).

G. Operations count

To provide an idea of the computational requirements of using Algorithm 4.1, the number of operations required in terms of the number of singular value \((\sigma)\), real perturbation value \((\tau)\), and eigenvalue \((\lambda)\) problems are noted in Table I.

It is to be noted that every time the radius is evaluated at a particular \(s \in \mathbb{C}\), a real perturbation value problem is solved. This occurs in step (1) and (2-f). In step (1), the real perturbation value problem is evaluated only once, but step (2-f) may require multiple evaluations, depending on the number of trials needed before \(r_k < r_{k-1}\) is obtained. However, it is found from experiment that the number of trials is typically small (i.e. close to 1 when using the method outlined in Section IV-B).

It is also to be noted that the main portion of the total operations count is from computing \(R_k^\sigma\) for all \(\theta \in S_k\) in step (2-c). In terms of implementation, \(S_k\) is discretized into a number of points. From experiment, it is found that it is generally sufficient to discretize \(S_k\) into 15 to 20 points, or a minimum of 0.5° (i.e. a maximum of 360 \(= \frac{\pi}{\frac{\pi}{2}}\) points), whichever results in a smaller angular division. For a given \(\theta \in S_k\), computing \(R_k^\sigma\) requires: i) solving for the eigenvalues, \(\Lambda\), of the \(2n \times 2n\) matrix \(\mathcal{H}(\tilde{r}_{k-1}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{T}_{k-1}^{-1} \tau_{k-1}, n+r, P_{nr}, P_{nm} T_{k-1}^{-1} n+m);\

and ii) evaluating a number of \(2(n+r) \times 2(n+m)\)-sized singular value problems to verify which real nonnegative eigenvalues, \(s \in \Lambda\), (there are a maximum of \(2n\) such values) have the property that they satisfy (18). Furthermore, since there are at most \(2n\) real nonnegative eigenvalues \(s \in \Lambda\) that may satisfy (18), \(R_k^\sigma\) contains at most \(2n+1\) intervals. Hence, to determine which intervals in \(R_k^\sigma\) achieve a radius less than \(r_{k-1}\), at most \(2n+1\) points are tested, resulting in \(2n+1\) additional \(2(n+r) \times 2(n+m)\)-sized singular value problems.

Remark 4.1: It should be noted that Algorithm 4.1 solves only a limited number of real perturbation value problems, which by itself is a 1-D optimization problem involving multiple \(2(n+r) \times 2(n+m)\)-sized singular value problems. However, this proves to be an advantage of Algorithm 4.1, as compared to say, a gradient search method, which requires solving a real perturbation value problem at each trial point, and at each step involved with computing the steepest gradient.

| V. NUMERICAL EXAMPLE |

The following example can be found in [2], where the real controllability radius of

\[
A = \begin{bmatrix} 1 & 1 & 1 \\ 0.1 & 3 & 5 \\ 0 & -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.1 \\ 0 \end{bmatrix}
\]

is found to be 0.0492, and is achieved at \(s = 0.972 + j0.982\).

Using Algorithm 4.1 with the (arbitrary) starting point \(s_0 = 1\), the same radius and global minimizer are found to within 3 significant figures in 5 iterations and to 6 significant figures in 7 iterations. Table II lists \(r_k, s_k\), and the “minimizing set”, \(S_k\) for each iteration \(k = 0, \ldots, 9\). Figure 1 provides a grid plot of the real controllability radius with respect to a given \(s \in \mathbb{C}\). Figures 2 to 5 plot \(R_k\) (outlined by the dots) obtained at Iterations 1, 3, 4, and 8 respectively, superimposed on a contour plot of Figure 1. The straight lines originating from the origin depict \(S_k\).

It is interesting to note that on a computer with a Pentium IV 2.0GHz processor, 512MB of RAM, and MATLAB 7.0, Algorithm 4.1 took a total of about 6 sec. to complete, which is approximately the same amount of time a gradient search method requires to obtain a (possibly local) minimum from a single starting point. Hence, if the gradient search method has to test 20 initial points in order to achieve a certain level of confidence that the obtained minimum is global, then the gradient search method will take approximately 20 times longer to run than Algorithm 4.1.

VI. CONCLUSIONS

In this paper, an efficient algorithm is presented for solving the general 2-D optimization problems (10) and (11), which are essential for computing the complex and
real controllability radius, the minimum-phase radius, the DFM radius, etc. The algorithm works by iteratively reducing the size of the so-called “minimizing set” which contains the global minimizer. Unlike methods such as the gradient search method, the choice of the initial point of the proposed algorithm is not crucial, and the minimum is obtained with a high degree of confidence that it is indeed global.

REFERENCES