On an Inviable Approach for Derivation of 2-D Stability Tests

Yuval Bistritz, *Fellow, IEEE*

***Abstract*—A tabular stability test for two-dimensional (2-D) discrete systems that was published in these Transaction is shown to be not correct. It is also shown that the claimed new method that it introduced to extend stability conditions from one-dimensional (1-D) to 2-D systems relies on a mathematically inviable argument. The paper tries to find a similar but correct algorithm and stability conditions. The outcome of the search after a stability test with similar algorithm is a variant of the Maria–Fahmy 2-D stability test for which a more concise set of necessary and sufficient conditions for stability are obtained. The search after stability conditions of similar appearance that can be posed on the correct algorithm, yields new necessary conditions for 2-D stability that resemble stability conditions associated with the "reflection coefficient" parameters in the 1-D Schur test.**

***Index Terms*—Continuous-discrete stability, multidimensional systems, stability tests, two-dimensional (2-D) discrete stability.**

I. Introduction

STABILITY testing of two-dimensional (2-D) discrete systems requires a verification on whether a bivariate polynomial

$$D(z_1, z_2) = \sum_{i=0}^{n_1} \sum_{k=0}^{n_2} d_{i,k} z_1^i z_2^k \tag{1}$$

has no zeros in the closed exterior of the unit bi-circle, viz.,

$$D(z_1, z_2) \neq 0 \qquad \forall (z_1, z_2) \in \bar{V} \times \bar{V} \tag{2}$$

where $\bar{V} = V \cup T$ with $T = \{z : |z| = 1\}$ and $V = \{z : |z| > 1\}$. A polynomial for which the condition in (2) holds will be called 2-D stable.

The paper will show that a method proposed to test the above condition in [1] is not correct. The paper also investigates how this incorrect result was reached. It is shown that the claimed new approach to extend stability conditions from one- to 2-D systems relies on an inviable mathematical argument. Pointing on fault in the underlying approach has a broader impact in that it may save further futile research and in disqualifying additional already published papers that used the incorrect approach. One such instance that we noticed is the continuous–discrete stability test in [2].

A second undertaking of this paper is to find a similar but correct 2-D stability algorithm and stability conditions to what was attempted in [1]. This route will lead to a couple of new results in 2-D stability as follows. Looking after a 2-D stability test that uses a more or less similar recursion will lead to a refined version for the first published tabular 2-D stability test—the Maria–Fahmy test [3]. While searching for stability conditions that can be posed on the algorithm and have a similar appearance to those suggested in [1] will lead to a some interesting new necessary conditions for 2-D stability that are related to conditions that hold for the "reflection coefficients" in the Schur one-dimensional (1-D) stability test.

The paper is constructed as follows. The next section introduces our notation and uses it to rephrase the method that was proposed in [1]. Section III derives a correct similar algorithm for testing 2-D stability. Section IV compares the two procedures and identifies at what point and why the proof of the main results in [1] is not viable. Finally, Section V derives the aforementioned additional necessary conditions for 2-D stability.

II. Preliminaries

The form used in (1) and (2) to define 2-D stability is not the only conventions that has been used in the literature on 2-D stability. The paper [1] uses one of the alternative forms as follows. A bivariate polynomial

$$B(z_1, z_2) = \sum_{i=0}^{n_1} \sum_{k=0}^{n_2} b_{i,k} z_1^i z_2^k \tag{3}$$

is tested to whether it satisfies the condition of no zeros in the closed interior of the unit bi-circle, viz.,

$$B(z_1, z_2) \neq 0 \qquad \forall (z_1, z_2) \in \bar{U} \times \bar{U} \tag{4}$$

where $\bar{U} = U \cup T$ with $U = \{z : |z| < 1\}$. To avoid ambiguity, a polynomial (3) for which the condition (4) holds will be termed (2-D) anti-stable.

We used the "stable" convention (2) in all our previous papers on 2-D stability, e.g., [4] and [5]. We shall mostly adhere to this convention also here. It is the most convenient convention to extend extension of 1-D stability conditions to 2-D stability conditions because it is consistent with regarding a (1-D) discrete-time system as stable when its characteristic polynomial $p(z) \neq 0 \quad \forall z \in \bar{V}$ (has all its zeros in U). A univariate polynomial that has all its zeros in U or V will be called (1-D) stable or anti-stable, respectively. The transition between the two conventions involves a simple reversion of rows and columns of the polynomial coefficients (an operation to be defined in a moment). Translation of algorithms and conditions from their "anti-stable" convention in [1] to "stable" terms will be provided to facilitate comparisons.

The remaining of our notation and terminology can be summarized as follows. A polynomial like (1) is said to have degree (n_1, n_2). A same letter is used to denote both the polynomial and

Manuscript received April 22, 2003; revised August 1, 2004. This paper was recommended by Associate Editor J. Vandewalle.

The author is with Department of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: bistritz@eng.tau.ac.il).

Digital Object Identifier 10.1109/TCSII.2005.852929

its coefficient matrix (or vector). Thus, the above bivariate polynomial may be written also as $D(z_1,z_2)=\mathbf{z_1}^t D\mathbf{z}_2$, with $\mathbf{z}:=[1,z,\ldots,z^i,\ldots]^t$ of length depending on context. Similarly a univariate polynomial can be written as $p(z)=\mathbf{z}^t p$. We denote $p^r=Jp$, where J is the reversion matrix (a square matrix with 1's on its anti-diagonal and zeros elsewhere) and $p^\sharp = Jp^\star$, where * denotes complex conjugate. Correspondingly, the next polynomials are defined, $p^r(z)=\mathbf{z}p^r=z^n p(z^{-1})$ and $p^\sharp(z)=\mathbf{z}^t p^\sharp = z^n p^\star(z^{-1})$. Similarly, for matrices and bivariate polynomials we denote $D^r=JDJ$ and $D^\sharp=JD^\star J$ and correspondingly have $D^r(z_1,z_2)=\mathbf{z_1}^t D^r\mathbf{z_2}=z_1^{n_1}z_2^{n_1}D(z_1^{-1},z_2^{-2})$ and $D^\sharp(z_1,z_2)=\mathbf{z_1}^t D^\sharp\mathbf{z}_2=z_1^{n_1}z_2^{n_1}D^\star(z_1^{-1},z_2^{-2})$.

It becomes apparent that $D(z_1,z_2)$ is stable if, and only if, a $B(z_1,z_2)$ with coefficient matrix related to it by $B=D^\sharp$ (or $B=D^r$) is anti-stable. Similarly, $p(z)$ is stable if, and only if, $p^\sharp(z)$ (or $p^r(z)$) is anti-stable.

The method proposed in [1] to test 2-D stability consists of an algorithm to build a table whose entries are univariate polynomials and from stability conditions posed on this table. The following Algorithm A duplicates the construction rules for [1, Table I] as a recursion for bivariate polynomials.

Algorithm A [The Algorithm in [1] Arranged Into a Polynomial Recursion]: Consider a $B(z_1,z_2)$ in (3) such that $B(0,0) > 0$ and express it as $B(s,z) = \sum_{k=0}^n b_{0,k}(s)z^k, n := n_2$. Assign to it a sequence of polynomials, $B_i(s,z)=\sum_{k=0}^m b_{[i]k}(s)z^k, i=1,\ldots,n$, by the following recursions.

For $i=1,\ldots,n-1$ do

$$zB_i(s,z)=b_{[i-1]0}(s)B_{i-1}(s,z)-b_{[i-1]n-i+1}(s)B_{i-1}^r(s,z). \quad (5)$$

The next Assertion A appears in [1, Th. 3].

Assertion A. [Main Result in [1], Not Correct]: Assume Algorithm A is applied to a $B(s,z)$ with $B(0,0)>0$. The condition (4) holds for $B(s,z)$ if, and only if, the following set of conditions hold.

1) $b_{[i],0}(1)\pm b_{[i]n-i}(1)>0$ for $i=0,\ldots n-1$.
2) $b_{[i],0}(-1)\pm b_{[i]n-i}(-1)>0$ for $i=0,\ldots n-1$.
3) $b_{[i]0}(s)\pm b_{[i]n-i}(s)\neq 0\ \forall z\in\bar{U}$ (are anti-stable) for $i=0,\ldots,n-1$.

We shall later (in Section IV) convert Algorithm A and Assertion A from its above "anti-stable" setting to "stable" convention terms which will simplify their comparison with a proper 2-D stability test of close form that will be derived in Section III.

We shall need the next Lemma for our derivation.

Lemma 1: $D(z_1,z_2)$ is stable if, and only if the following are true.

1) $D(z,a)\neq 0$ for all $z\in\bar{V}$ and some $a\in\bar{V}$. (6)
2) $D(s,z)\neq 0$ for all $(s,z)\in T\times\bar{V}$. (7)

The above lemma, cited also in [1] (as Theorem 2 there) has been indeed the starting point of essentially all methods for testing 2-D stability. It was introduced to this field by Huang [6] (with $a=\infty$) and extended to the above form by Strintzis [7].

In motivating their new approach, the authors of [1] count two drawbacks in using Lemma 1: one is that it leads to the occurrence of complex polynomials; second—that at the end the resulting 2-D stability tests require testing of several univariate polynomials for a positivity condition. The new approach proposed instead a 2-D tabular test on which the 2-D stability conditions appear as 1-D stability conditions that real polynomials extracted from the table have to satisfy.

The first assumed drawback is not precise. The lemma lead to tabular tests with only real polynomials and arithmetics when D is real (and complex coefficients if D is complex e.g., [4]). This will be apparent also from our derivation in this paper that too starts with Lemma 1. The second statement is essentially correct. However, the testing of positivity conditions on T has the same complexity as testing 1-D stability [8]. Also, one positivity test is enough. Incidently, the wish to carry out 2-D stability by a collection of real 1-D stability tests that was expressed in the conclusion of [1] has been attained recently [9], not as a collection of real 1-D stability tests that are posed on a 2-D stability table but in a more fulfilling manner—a collection of 1-D stability tests that *replace* the construction of the 2-D stability table.

The approach in [1] started with a certain simplification of stability conditions posed on the Inner (or Schur–Cohn) determinants that is possible when the tested polynomial is real [10]. It was assumed there that using stability conditions for real polynomials will circumvent complex polynomials in the 2-D stability test. A major difficulty that the paper will track is that the manner the paper extended the 1-D stability conditions to bivariate polynomials makes incorrect mathematical assumptions. Irrespectively, a secondary difficulty that will also be revealed is that the relations between Algorithm A and the Inners determinants that were assumed are not correct (not even for real polynomials).

III. A Comparable Proper 2-D Stability Test

The Schur procedure to test 1-D stability have appeared in the literature in many forms since it was introduced (more than 80 years ago). Its offsprings consist typically of an algorithm that builds for the tested polynomial a sequence of polynomials of descending degrees (often presented by a table whose rows are the polynomials coefficients) and from stability conditions posed on the coefficients of the produced polynomials. The many tests that followed from the Schur procedure can be classified into four types, labeled from "A" to "D" in [11]. The D-type is the best candidate to obtain an algorithm similar to Algorithm A because it is the only one among the four types that like Algorithm A involves no division. Our starting point is therefore the prototype algorithm for D-type 1-D stability tests in [11] that is cited below as Algorithm 1 and Theorem 1.

Algorithm 1: Construct for a polynomial $p(z)=\sum_{i=0}^n p_i z^i$ (with complex coefficients) a sequence of polynomials $f_m(z)=\sum_{i=0}^n f_{m,i}z^i, m=n-1,\ldots 0$ as follows. Set $f_n(z):=p(z)$ then for $m=n,\ldots,1$ do

$$zf_{m-1}(z)=f_{m,m}^\star f_m(z)-f_{m,0}f_m^\sharp(z),\quad f_n(z)=p(z). \quad (8)$$

Alternatively, since all $f_{m,m}$ are real for $m\le n-1$, the algorithm may also written as follows:

$$\begin{aligned} zf_{n-1}(z)&=p_n^\star p(z)-p_0 p^\sharp(z)\\ zf_{m-1}(z)&=f_{m,m}f_m(z)-f_{m,0}f_m^\sharp(z),\\ &\qquad m=n-1,\ldots,1. \end{aligned} \quad (9)$$

Theorem 1 [1-D Stability Conditions for Algorithm 1: $p(z)$ is stable if, and only if, Algorithm 1 produces for it

$$f_{m,m} > 0, \qquad m = n-1, \ldots, 0. \tag{10}$$

Proof of Theorem 1 can be deduced from the zero location rules proved for Algorithm 1 in [11] (which, by the way, has a tricky shape quite unsuspected considering the above simple stability conditions). A direct and simple proof for Theorem 1 follows from stability conditions posed on the so called "reflection coefficients" (once the latter are taken to be known). This short proof is brought next (also because it introduces equations that will be used later in Section V to obtain some new 2-D stability condition). The reflection coefficients are parameters associated with $p(z)$ that can be obtained from the sequence that Algorithm 1 produces as follows:

$$k_m = -\frac{f_{m,0}}{f^{\star}_{m,m}}, \qquad m = 1, \ldots, n. \tag{11}$$

It is well known that $p(z)$ is stable if, and only if

$$|k_m| < 1, \qquad m = 1, \ldots, n. \tag{12}$$

Various proofs are available for the above assertion (including one in [11]). A Proof for Theorem 1 follows readily from using the 1-D stability conditions (12) with the next relation

$$\begin{aligned} f_{n-1,n-1} &= |p_n|^2 - |p_0|^2 = |p_n|^2(1 - |k_n|^2) \\ f_{m-1,m-1} &= f^2_{m,m} - |f_{m,0}|^2 = f^2_{m,m}(1 - |k_m|^2), \\ &\qquad m = n-1, \ldots 1 \end{aligned} \tag{13}$$

that are obtained by comparing leading coefficients in (9).

To pursue a 2-D stability test based on Lemma 1 and Algorithm and Theorem 1, it is useful to replace the second condition of Lemma 1 $D(s,z)$ with $D(\tilde{s},z) := s^{-n_1/2}D(s,z)$ and regard $D(\tilde{s},z)$ as a univariate polynomial in z with coefficients that are "balanced polynomials" dependent on $s \in T$, $D(\tilde{s},z) = \sum_{k=0}^{n} d_k(\tilde{s})z^k$. (A balanced polynomial $q(\tilde{s})$ is a polynomial in the two variables s and s^{-1} or $s^{1/2}$ and $s^{-1/2}$ related to the univariate polynomial $q(s)$ of degree n by $q(\tilde{s}) = s^{-n/2}q(s)$. Changing $D(s,z)$ with $D(\tilde{s},z)$ is legitimate because clearly the condition (7) holds if, and only if, "$D(\tilde{s},z) \neq 0$ for all $(s,z) \in T \times \bar{V}$". The advantage of balanced polynomial is that for $s \in T, (q(\tilde{s}))^{\star} = q^{\sharp}(\tilde{s}))$. As a consequence, we can test the second condition in Lemma 1 by the applying the above 1-D stability test to the univariate polynomial $p_s(z) = D(\tilde{s},z)$ (with balanced polynomial coefficients). The next algorithm is readily obtained from Algorithm 1 (it can be realized by formal replacement of each complex conjugate operation with the reversion operation).

Algorithm 2: Assign to the tested polynomial, written as $D(s,z) = \sum_{k=0}^{n} d_k(s)z^k, n := n_2$, a sequence of polynomials $\{F_m(s,z) = \sum_{k=0}^{m} f_{[m]k}(s)z^k, m = n-1, \ldots, 0\}$, using the following recursions:

$$zF_{n-1}(\tilde{s},z) = d_n^{\sharp}(\tilde{s})D(\tilde{s},z) - d_0(\tilde{s})D^{\sharp}(\tilde{s},z). \tag{14a}$$

For $m = n-1, \ldots, 1$ do

$$zF_{m-1}(\tilde{s},z) = f_{[m]m}(\tilde{s})F_m(\tilde{s},z) - f_{[m]0}(\tilde{s})F_m^{\sharp}(\tilde{s},z). \tag{14b}$$

After Algorithm 2 has been established as suggested, it becomes apparent that replacing everywhere $\tilde{s}$ by s does not affect the sequence of coefficient matrices that the algorithm produces. This follows because powers of s that present the difference between balanced and normal polynomials of s always cancels out between the two sides of the (14). For example, in the first step of the recursion

$$\begin{aligned} zF_{n-1}(\tilde{s},z) &= s^{-n_1}zF_{n-1}(s,z) \\ &= s^{-n_1/2}d_n^{\sharp}(s)s^{-n_1/2}D(s,z) \\ &\quad - s^{-n_1/2}d_0(s)s^{-n_1/2}D^{\sharp}(s,z) \end{aligned}$$

and so forth.

From the manner Algorithm 2 was obtained from Algorithm 1 it follows that Theorem 1 in conjunction with Lemma 1 implies the following conditions for 2-D stability.

Theorem 2 [2-D Stability Conditions for Algorithm 2]: $D(z_1,z_2)$ is stable if, and only if, the following set of conditions hold.

1) $D(z,1) \neq 0 \; \forall z \in \bar{V}$.
2) $D(1,z) \neq 0 \; \forall z \in \bar{V}$.
3) $f_{[m]m}(s) \neq 0 \; \forall s \in T$ (or $f_{[m]m}(\tilde{s}) > 0 \; \forall s \in T$ for $m = n-1, \ldots, 0$) where $f_{[m]m}(s)$ are produced by applying Algorithm 2 to $D(z_1,z_2)$.

The next theorem reveals that the above set of necessary and sufficient conditions for 2-D stability is too large.

Theorem 3 [Tighter 2-D Stability Conditions for Algorithm 2: $D(z_1,z_2)$ is stable if, and only if, the following set of conditions hold.

1) $D(z,1) \neq 0 \; \forall z \in \bar{V}$.
2) $D(1,z) \neq 0 \; \forall z \in \bar{V}$.
3) $F_0(s,z) = f_{[0]0}(s) \neq 0 \; \forall s \in T$ (or $f_{[0]0}(s) > 0 \quad s \in T$) where $F_0(s,z) = f_{[0]0}(s)$ is the last polynomial that has degree 0 in z (and degree $2^{n_2}n_1$ in s) that Algorithm 2 produces for $D(z_1,z_2)$.

Proof: The conditions are necessary for stability as a subset of the conditions in Theorem 2. To prove sufficiency we prove that the conditions 1)–3) here imply the larger set in Theorem 2. Assume they hold and nevertheless the conditions in Theorem 2 do not hold. That means that some of the $f_{m,m}(s)$ may have zeros on T. Consider then the balanced form of the algorithm (i.e., with $\tilde{s}$ replacing s everywhere). Then, $f_{m,m}(\tilde{s})$ are real for all $m < n$. Say i is the highest index of balanced polynomial that vanishes on T. Condition 2) implies that at $s = 1 f_{m,m}(\tilde{s}) > 0$ for all $m < n$. Let then s_i be the rightmost zero (i.e., the zero on T closest to $s = 1$), $f_{i,i}(\tilde{s}_i) = 0$. It follows from the relations

$$f_{[m-1],m-1}(\tilde{s}) = f_{[m]m}(\tilde{s})^2 - |f_{m,0}(\tilde{s})|^2$$

[it is the counterpart of (13)] that $f_{[i-1],i-1}(\tilde{s}_i) \leq 0$. Let then s_{i-1} be the rightmost zero on T of $f_{[i-1],i-1}(\tilde{s})$. (Note that it will be closer to $s = 1$, i.e., the $s_i^R \leq s_{i-1}^R < 1$ where s^R denotes real part of s.) Then again one has $f_{[i-2],i-2}(\tilde{s}_{i-1}) \leq 0$. Therefore, $f_{[i-2],i-2}(\tilde{s})$ must vanish on the arc of T between s_i and $s = 1$. Keep repeating this argument leads to that $f_{[0]0}(s_0) = 0$ for some s_0 on T with $s_i^R \leq s_{i-1}^R \leq \ldots \leq s_0^R < 1$. A contradiction to the given condition 3). ■

Example: Here is a simple numerical illustration for the above 2-D stability test (Algorithm 2 and Theorem 3). This example will later serve also as a counter example to the validity of the method in [1]. Consider the next polynomial of degree $(1,2)$

$$D = [1, z_1] \begin{bmatrix} 1 & 5 & 7 \\ 2 & 3 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ z_2 \\ z_2^2 \end{bmatrix}.$$

The polynomial is not 2-D stable. To furnish this fact it is enough to provide a zero $D(s_o, z_o) = 0$ with $|s_o| > 1, |z_o| > 1$. Such a zero occurs at (approximately) $s_o = -0.99000 + 0.20000j$ and $z_o = 0.20033 + j1.02404$.

Testing this polynomial with Algorithm 2 and Theorem 3 proceeds as follows. First $D(z,1) = 13 + 14z$ is evidently 1-D stable. Similarly, $D(1,z) = 3 + 8z + 16z^2$ can also be determined to be 1-D stable. Next, Algorithm 2 is performed. It produces

$$F_1 = \begin{bmatrix} 42 & 61 \\ 51 & 125 \\ 11 & 61 \end{bmatrix} \quad F_0 = \begin{bmatrix} 3259 \\ 12\,547 \\ 18\,581 \\ 12\,547 \\ 3259 \end{bmatrix}.$$

It remains to check whether $\mathbf{s}^t F_0 \neq 0 \; \forall s \in T$. This condition is false (its 4 zeros are all on the unit-circle). Therefore, the test determines $D(z_1, z_2)$ as not 2-D stable.

Algorithm 2 with Theorem 2 are essentially a replica of the first 2-D stability test that was proposed by Maria and Fahmy in [3]. The single positivity test in Theorem 3 is new for this test. It can be deduced from (14) that the polynomials that the algorithm produces are of degree $F_{n-k}(s,z)$ has degree $(n_1 2^k, k)$ for $k = 1, \ldots n = n_2$. As a result, the overall cost of computation has an even more severe exponential growth with the degree of the tested polynomial (it contain $n_1^2 2^{2n_2}$ as its most dominant term). This makes this test by far less efficient than second generation tabular tests like [12], [4], [5] that have only $O(n^6)$ complexity [for a polynomial of degree (n,n)] or the more recent third generation tests that were obtained as their interpolated counterparts in [5] and [13] that require only $O(n^4)$ operations.

IV. Comparison With [1]

This section first shows that the previous numerical illustration provides a counterexample to the validity of the method proposed in [1]. Then, a careful comparison of the method in [1] with the 2-D stability test in the previous section is held to reveal difficulties in the manner of derivation in [1]. Finally, the question of whether (and to what extent) can the stability condition in the previous section be brought to terms with the conditions attempted in [1] is addressed.

Counter Example: Consider again the numerical example in the previous section. The method in [1] would test the polynomial $B(z_1, z_2) = D^r(z_1, z_2)$

$$B(z_1, z_2) = [1, z_1] \begin{bmatrix} 9 & 3 & 2 \\ 7 & 5 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ z_2 \\ z_2^2 \end{bmatrix}$$

through the following steps. The degree of this polynomial was chosen to be equal to the degree of a polynomial worked out in detail in Example 1 in [1] in order to facilitate the comparison of the subsequent steps here with respective steps detailed in Example 1 there.

Algorithm A starts with $B_0(s,z) = B(s,z)$. The next bivariate polynomial it produces is

$$B_1(s,z) = [1, s, s^2] \begin{bmatrix} 77 & 21 \\ 122 & 53 \\ 48 & 30 \end{bmatrix} \begin{bmatrix} 1 \\ z \end{bmatrix}.$$

Now we turn to examine the conditions in Assertion A. The polynomials used to build the polynomials required in Assertion A are $b_{[0]0}(s) = 9 + 7s, b_{[0]2}(s) = 2 + s$ obtained from $B_0(s,z)$ and $b_{[1]0}(s) = 77 + 122s + 48s^2$, and $b_{[1]1}(s) = 21 + 53s + 30s^2$ taken from $B_1(s,z)$. Using them one obtains $b_{[0]0}(s) + b_{[0]2}(s) = 11 + 8s$ that is positive at both ± 1 and is anti-stable, and $b_{[0]0}(s) - b_{[0]2}(s) = 7 + 6s$ that is too positive at both ± 1 and is anti-stable. Next, $b_{[1]0}(s) + b_{[1]1}(s) = 98 + 175s + 78s^2$ and $b_{[1]0}(s) - b_{[1]1}(s) = 56 + 69s + 18s^2$ are again both positive at both ± 1. (In principle, Algorithm A can be continued for one more step and produce $B_2(s,z) = 4588 + 16\,562s + 18\,207s^2 + 8532s^3 + 1404s^4$ that is too anti-stable, but this last $i = n$ step was not requested in [1].) Since all the sufficient conditions in Assertion A (i.e., in [1, Th. 3]) hold, this theorem implies that $B(z_1, z_2)$ is anti-stable (has no zeros in the closed interior of the bi-disk). This conclusion is in disagreement with the conclusion from Algorithm 2 and Theorem 3 that determined for this numerical example that $D(s,z) = B^r(s,z)$ is not 2-D stable. Using the reciprocal of the point (s_o, z_o) with $|s_o| > 1, |z_o| > 1$ which in the previous numerical illustration where shown as a zero of $D(s,z)$ proves that $B(1/s_o, (1/z_o)) = 0$ at a point in the interior of the bi-disk. Thus, the above is a counter-example to the validity of the method proposed in [1].

The proof for [1, Th. 3] (i.e., Assertion A) uses another theorem (Theorem 4) there that employs 1-D stability conditions that Jury posed on Inner determinants [10]. The Inner is a matrix of size $2n \times 2n$ that can be assigned to a polynomial of degree n such that its 1-D stability can be determined from positivity of the determinants of a sequence of n submatrices of sizes $2k \times 2k$ ordered in an inner-wise nested manner. It was advanced by Jury as an alternative to the resultant matrix (again a $2n \times 2n$ matrix with differently arranged for quadrants). The Inners and the resultants contain the same information as the unit-circle Bezoutian, also known as the Schur–Cohn or Schur–Cohn–Fujiwara (SCF) matrix [10], [14]. The SCF matrix is of size $n \times n$ only and features the property that its sequence of principal minors is equal to the values of the sequence of determinants of the inner submatrices (choosing a consistent assignment, else the two sequences may differ by ± 1 signs and scaling that runs along the sequence). Necessary and sufficient conditions for 1-D stability is positive definiteness of the SCF matrix.

Using the SCF matrix instead of Inners admits an explanation of why the Proof in [1] is not viable in terms of basic knowledge of algebra of matrices. The SCF matrix is a Hermitian complex matrix for a complex univariate polynomial (and a real symmetric matrix for a real polynomial). Positive definiteness is equivalent to the condition that all the determinants of its main submatrices (all the principal minors) are real and positive. This condition is also equivalent to that all its eigenvalues being real and positive. Thus, in particular real eigenvalues are necessary conditions for a matrix to be positive definite. It is also clear that nonsingularity of the matrix (i.e., no zero eigenvalues) is *not* a sufficient condition for positive definiteness.

The derivation in [1] uses a simplified setting of the conditions posed on Inners that Jury showed that can be used for real (as opposed to complex) coefficient univariate polynomials. The proof there proceeds by regarding the polynomial $B(s,z) = \sum_{k=0}^{n} b_{0,k}(s)z^k$ in Algorithm A as a univariate polynomial in the indeterminate z whose coefficients are the polynomials $b_{0,k}(s)$. It then sets these coefficients into the simplified Inner matrix for real polynomials and uses some sort of continuity argument on what happens to the eigenvalues as the value of the polynomials $b_{0,k}(s)$ vary. In corresponding terms, one assigns to the above univariate polynomial a real SCF matrix (with polynomial entries) and examines what happens as $b_{0,k}(s)$ vary. As long as $b_{0,k}(s)$ take real values (as long as s is real) the corresponding SCF matrix will have real eigenvalues. However, when $b_{0,k}(s)$ take complex values, insisting on using the real SCF (that, amounts in using Jury's simplified Inners for a real polynomial) the assigned SCF matrix becomes a complex matrix that is symmetric but not Hermitian. (Its lower triangular is the transpose but not the conjugate transpose of its upper triangular.) The argument in [1] that concentrates on watching whether the eigenvalues vanish or not as the coefficients vary becomes meaningless because for complex entries the eigenvalues are no longer guaranteed to remain real. Consequently, they truly also cease to provide any information on the location of the zeros of the examined complex univariate polynomial.

To facilitate comparison of the method in [1] with the method that was obtained in Section III it is possible to eliminate artificial differences by translating Algorithm A and assertion A from the "anti-stable" to the "stable" convention. Using the following substitutions:

$$\begin{aligned} B(z_1,z_2) &= D^r(z_1,z_2) \\ A_i(z_1,z_2) &= B^r_{n-i}(z_1,z_2), \qquad i=0,\ldots,n. \end{aligned} \tag{15}$$

Algorithm A and Assertion A take the next forms.

Algorithm $\hat{A}$: Consider a $D(z_1,z_2)$ such that $d_{n_1,n_2} > 0$ and express it as $D(s,z) = \sum_{k=0}^{n} d_k(s)z^k, n := n_2$. Assign to it a sequence of polynomials $\{A_m(s,z) = \sum_{k=0}^{m} a_{[m]k}(s)z^k, m = n-1,\ldots,0\}$, using the following recursions:

$$zA_{n-1}(s,z) = d_n(s)D(s,z) - d_0(s)D^r(s,z). \tag{16a}$$

For $m = n-1,\ldots,1$ do

$$zA_{m-1}(s,z) = a_{[m]m}(s)A_m(s,z) - a_{[m]0}(s)A^r_m(s,z). \tag{16b}$$

Assertion $\hat{A}$ (Not Correct): Assume algorithm $\hat{A}$ is applied to $D(s,z)$ with $d_{n,n} > 0$. $D(s,z)$ is stable if, and only if, the following set of conditions hold.

1) $D(1,z) \neq 0 \; \forall z \in \bar{V}$.
2) $D(-1,z) \neq 0 \; \forall z \in \bar{V}$.
3) $a_{[m]m}(s) \pm a_{[m]0}(s) \neq 0 \; \forall s \in \bar{V}$ (are both stable) for $m = n,\ldots,1$.

In stating Assertion $\hat{A}$ we substituted the translation of first two conditions in [1] (or Assertion A) by more concise equivalent conditions as follows. Direct translation of condition 1) in Assertion A gives $a_{[i],i}(1) \pm a_{[i]0}(1) > 0$ for $i = 0,\ldots n-1$. It therefore amounts to that all the reflection coefficients $k_{i+1} = (a_{[i]0}(1)/a_{[i],i})(1)$ of the real polynomial $D(1,z)$ have moduli less than one which holds if and only if $D(1,z)$ is stable, see (12). [Thus, condition 1) in Assertion A is equivalent to "$B(1,z)$ is anti stable".] By a similar justification, condition 2) was replaced by "$D(-1,z)$ is stable" (and in Assertion A a corresponding "$B(-1,z)$ is anti-stable" is possible).

Note that each instance that Algorithm 2 has the sharp operator $^\sharp$, Algorithm $\hat{A}$ has instead only the reversion operator ^{r}. Recall that Algorithm 2 has been obtained from Algorithm 1 regarding the tested polynomial as a univariate polynomial whose coefficients are *complex* valued 'balanced polynomials' for indeterminate variable $s \in T$. Algorithm $\hat{A}$ is therefore similar to Algorithm 1 when the polynomial coefficients take real values.

There is a secondary layer of difficulty in [1]. It can be shown that [1, Th. 3] assumes incorrect relations between the Inner determinants and the first and last coefficient $b_{i,k}$ of $B_i(s,z)$ in Algorithm A. The relations there are not correct even for a polynomial with simple (not polynomial) real coefficients. The correct relations can be worked out from relations in [11] between the leading coefficients in the sequence of polynomials produced by C-type prototype test there (which are equal to the principal minors of the Schur–Cohn matrix and therefore are also equal to the inner determinants) and the leading coefficients in the D-type test there (cited here as Algorithm 1 and Theorem 1). A quicker but sufficient evidence that the relations in [1, Th. 3] can not be true, can be drawn from the observation that there is inconsistency in the degrees of the polynomials in the relation

$$\det[X_i(z_1) \pm Y_i(z_1)] = b_{i,0}(z_1) \pm b_{i,n_2-i}(z_1), \qquad i = 0,\ldots,n_2-1$$

between the two sides of (12) and (13) in [1]. At the left-hand side the ith polynomial has degree $(i+1)n_1$ because it presents the determinant of a polynomial matrix of size $(i+1)\times(i+1)$ with entries that are polynomials of degree n_1; At the right-hand side, the polynomial has degree $2^i n_1$ (equal to the degree in s of $A_{n_2-i}(s,z)$ and $F_{n_2-i}(s,z)$).

V. Further Necessary Conditions for 2-D Stability

The last question that will be addressed is: what stability conditions, if any, can be attached to polynomials that look like the polynomials $a_{[m]m}(z_1) \pm a_{[m]0}$ in [1] (i.e., in Assertion $\hat{A}$ here). Thus, we seek relation between 2-D stability and properties of polynomials $f_{[m]m}(s) + f_{[m]0}(s)$ and $f_{[m]m}(s) - f_{[m]0}(s), m = n,\ldots,0$, where $f_{[m]m}(s)$ and $f_{[m]0}(s)$ are created by Algorithm 2.

Applying the conditions (13) to $D(\tilde{s},z) = \sum_{k=0}^{n} d_k(\tilde{s})z^k$, regarded as a univariate polynomial with 'balanced polynomial' coefficients dependent on $s \in T$, imply the conditions

$$\begin{aligned} d_n(\tilde{s})d_n^\sharp(\tilde{s}) - d_0(\tilde{s})d_0^\sharp(\tilde{s}) &> 0 \\ f^2_{[m]m}(\tilde{s}) - f_{[m]0}(\tilde{s})f^\sharp_{[m]0}(\tilde{s}) &> 0, \qquad m = n-1,\ldots,1. \end{aligned} \tag{17}$$

It is possible to relax the above conditions and replace balanced polynomials with normal polynomials (dependency on $\tilde{s}$ by dependency on s). As a consequence, it is possible to replace condition 3) in Theorem 2 by

$$d_n(s)d_n^\sharp(s) - d_0(s)d_0^\sharp(s) \neq 0 \qquad \forall s \in T \tag{18}$$

$$f_{[m]m}^2(s) - f_{[m]0}(s)f_{[m]0}^\sharp(s) \neq 0 \qquad \forall s \in T, \quad m = n-1, \ldots, 1. \tag{19}$$

Since these are symmetric polynomials of even degree, the no zeros on T condition amounts to having half of the zeros in U and half in V. Next, it follows from (11), (12) that for $s \in T$

$$|d_n(s)| > |d_0(s)| \qquad \forall s \in T$$

$$|f_{[m]m}(s)| > |f_{[m]0}(s)| \qquad \forall s \in T, \quad m = n-1, \ldots, 1.$$

Therefore, conditions (18)–(19) imply the conditions

$$d_n(s) \pm d_0(s) \neq 0 \qquad \forall s \in T \tag{20}$$

$$f_{[m]m}(s) \pm f_{[m]0}(s) \neq 0 \qquad \forall s \in T, \quad m = n-1, \ldots, 1. \tag{21}$$

Thus, it can also be said (using Rouche's theorem) that the polynomials in (21) have half of their zeros in U and half in V. We summarize these findings as the next theorem.

Theorem 4 [More Stability Conditions for Algorithm 2]:

1) $D(z_1, z_2)$ is stable if, and only if, the conditions 1) and 2) of Theorem 3 and $\hat{3}$): the conditions in (18)–(19) hold.
2) The conditions in (20) and (21) as well as the stronger condition that the polynomials in (21) have half of their zeros in U and the other half in V are necessary conditions for stability of $D(z_1, z_2)$.

With (20) and (21), we reached polynomials of a form that corresponds to those in condition 3) of Assertion $\hat{A}$. The condition (20) and that half of the zeros of the polynomials in (21) are in U and half in V (and no zeros on T) present necessary conditions for 2-D stability but not sufficient for stability because they do not imply those in (18) and (19). Therefore, in difference from (19), these conditions are *not* a sufficient set of conditions for 2-D stability not even in combination with conditions 1) and 2). Therefore, they can not substitute conditions 3) in Theorem 3. In particular, it is apparent that 1-D stability of the polynomials in (20) and (21) [like in condition 3) of Assertion $\hat{A}$] is neither necessary nor sufficient for 2-D stability. At the same time, testing zero location of a polynomial with respect to the unit circle (in order to determine how many are in V and in U or on T) need not be more complicated than testing whether it is 1-D stable [8].

VI. Conclusion

The paper showed that the method to test 2-D stability that was proposed in [1] is incorrect. The paper also identified that the new approach that was used there to extend stability conditions from 1-D to 2-D systems is not correct. Unfortunately, it seems that this incorrect approach was already used is several other cases. One instance is a method to test continuous-discrete stability conditions in [2]. Additional papers (that we have not seen) referenced in [1] and [2] that deal with related issues and are authored or co-authored by the same author(s) might also be afflicted. After the review of this paper was completed we discovered another paper in this transaction [15] that proposes to improve the algorithm in [1]. The counterexample in this paper can be used to verify that the stability condition there are too not correct.

The search after similar but correct 2-D stability test and stability conditions led to a couple of new results. One is an improved form for the Maria–Fahmy 2-D stability table [3] showing for it a tighter set of necessary and sufficient conditions for stability. The other is a new set of necessary condition for 2-D stability that hold for certain univariate polynomials that extend condition that hold for the "reflection coefficient" parameters in the Schur 1-D stability test.

References

[1] Y. Xiao and R. Unbehauen, "New stability test algorithm for two-dimensional digital filters," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 7, pp. 739–741, Jul. 1998.

[2] Y. Xiao, "Stability test for 2-D continuous-discrete systems," in *Proc. IEEE Conf. Decision Contr.*, 2001, pp. 3649–3654.

[3] G. A. Maria and M. M. Fahmy, "On the stability of two-dimensional digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, no. 4, pp. 470–472, Aug. 1973.

[4] Y. Bistritz, "Stability testing of 2-D discrete linear system polynomials by a 2-D tabular form," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 6, pp. 666–676, Jun. 1999.

[5] ——, "Stability testing of two-dimensional discrete-time systems by a scattering-type tabular form and its telepolation," *Multidim. Syst. Signal Process.*, vol. 13, pp. 55–77, Jan. 2002.

[6] T. S. Huang, "Stability of two-dimensional recursive filters," *IEEE Trans. Audio Electroacoustics*, vol. AU-20, pp. 158–163, Jun. 1972.

[7] M. G. Strintzis, "Test of stability of multidimensional filters," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 432–437, Aug. 1977.

[8] Y. Bistritz, "Zero location of polynomials with respect to the unit-circle unhampered by nonessential singularities," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 3, pp. 305–314, Mar. 2002.

[9] ——, "Testing stability of 2-D discrete systems by a set of real 1-D stability tests," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1312–1320, Jul. 2004.

[10] E. I. Jury, *Inners and the Stability of Linear Systems*. New York: Wiley, 1982.

[11] Y. Bistritz, "Reflection on schur-cohn matrices and jury-marden tables and classification of related unit circle zero location criteria," *Circuits Syst. Signal Process.*, vol. 15, no. 1, pp. 111–136, 1996.

[12] X. Hu and E. I. Jury, "On two-dimensional filter stability test," *IEEE Trans. Circuits Syst.*, vol. 41, no. 7, pp. 457–462, Jul. 1994.

[13] Y. Bistritz, "Stability testing of 2-D discrete linear systems by telepolation of an immittance-type tabular test," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 7, pp. 840–846, Jul. 2001.

[14] H. Lev-Ari, Y. Bistritz, and T. Kailath, "Generalized Bezoutians and families of efficient zero-location procedures," *IEEE Trans. Circuits Syst.*, vol. 38, no. 2, pp. 170–186, Feb. 1991.

[15] S.-F. Yang and C. Hwang, "An improved stability test algorithm for two-dimensional digital filters," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 7, pp. 1120–1123, Jul. 2000.