Circuits Systems Signal Process
Vol. 19, No. 3, 2000, Pp. 245–265

# Immittance-Type Tabular Stability Test for 2-D LSI Systems Based on a Zero Location Test for 1-D Complex Polynomials*

*Yuval Bistritz*[1]

**Abstract.** A new algebraic test is developed to determine whether or not a two-variable (2-D) characteristic polynomial of a recursive linear shift invariant (LSI, discrete-time) system is "stable" (i.e., it does not vanish in the closed exterior of the unit bi-circle). The method is based on the original form of a unit-circle zero location test for one variable (1-D) polynomials with complex coefficients proposed by the author. The test requires the construction of a "table", in the form of a sequence of centrosymmetric matrices or 2-D polynomials, that is obtained using a certain three-term recursion, and examination of the zero location with respect to the unit circle of a few associated 1-D polynomials. The minimal set necessary and sufficient conditions for 2-D stability involves one 1-D polynomial whose zeros must reside inside the unit circle (which may be examined before the table is constructed), and one symmetric 1-D polynomial (which becomes available after completing the table) that is required not to have zeros on the unit circle. A larger set of intermediate necessary conditions for stability (which may be examined during the table's construction) are also given. The test compares favorably with Jury's recently improved 2-D stability test in terms of complexity and munerical stability.

## 1. Introduction

An important consideration in the design and analysis of multidimensional linear discrete systems and filters is their stability. Two-dimensional (2-D) and higher-dimensional digital signals and systems arise in digital processing and modeling of images, waveforms from several sensors, and other applications: [11], [18].

The paper considers the problem of determining whether the 2-D polynomial

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

* Received June 16, 1999; revised March 22, 2000.

[1] Department of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: bistritz@eng.tau.ac.il

is stable, i.e., whether it satisfies the condition

$$D(z_1, z_2) \neq 0, \quad \text{for} \quad (z_1, z_2) \in \bar{V} \times \bar{V}, \tag{2}$$

where

$$T = \{z : |z| = 1\}, \qquad U = \{z : |z| < 1\}, \qquad V = \{z : |z| > 1\},$$

are used to denote the unit circle, its interior, and its exterior, respectively, and the bar denotes closure, $\bar{V} = V \cup T$.

A stable $D(z_1, z_2)$ is the key for stability of a discrete-time system described by a 2-D recursive difference equation. In a strict mathematical sense, a stable 2-D polynomial presents a sufficient condition for stability that becomes necessary only if the transfer function obtained by the $Z$-transform of the difference equation has no nonessential singularities of the second kind (NSSK) [12]. NSSK represents a peculiarity that is admitted in 2-D polynomials and has no counterpart in 1-D polynomials. Unlike 1-D polynomials, 2-D polynomials are not factorable in general. Consequently, two polynomials $N(z_1, z_2)$ and $D(z_1, z_2)$ that compose the transfer function $H(z_1, z_2) = N(z_1, z_2)/D(z_1, z_2)$ may be coprime and still admit $N(z_1, z_2)$ to have zeros on $T^2$ that coincide with zeros there of $D(z_1, z_2)$. NSSK corresponds to the possibility that at times such zeros may stabilize a system with an unstable $D(z_1, z_2)$. However, as pointed out in [20], a design of a filter is not likely to end with a $D(z_1, z_2)$ having NSSK zeros on $T^2$ and at the same time $D(z_1, z_2) \neq 0$ anywhere else in $\bar{V} \times \bar{V}$. In addition, such a situation may be argued not to present a system with acceptable robust stability for practical purposes. The fact that polynomials in more than one variable are not factorable is known to complicate the theory and design of multidimensional systems in additional ways. Most pertinent to stability testing is the fact that numerical calculation of the zeros (which for some applications is an adequate alternative to the algebraic stability test in the 1-D case) cannot be used to determine the stability of $D(z_1, z_2)$ (its set of zeros is not composed in general of a finite number of isolated zeros). More general discussions of stability of multidimensional systems are available in several texts, e.g., [20], [10], [11], [15].

This paper presents an *algebraic* 2-D stability test that aims to solve the problem in a finite count of operations. The method has been reported before in a conference paper [4], but without derivation details and proofs. The current paper brings all the derivation details and proofs required to establish the method. It also adds a certain simplification overlooked in the conference version that reduces by a factor of 4 the number of operations required to complete the test.

The test is based on the method for determining the distribution of the zeros of a 1-D polynomial with complex coefficients with respect to the unit circle in [3] that extends the real version in [2]. This 1-D test requires that the tested polynomial be real valued at $z = 1$, a requirement that may be fullfilled by a prescaling $p(z) \to p(1)^\star p(z)$. However, the implementation of this scaling in the context of applying the algorithm to 2-D stability testing causes a doubling of the row sizes of the initial matrices and consequently may quadruple the total cost of computation. A related alternative immittance-type 2-D tabular test in [7] does not encounter this problem because it is based instead on a modified form of the 1-D test [6] that does not need prescaling. The current paper incorporates an added measure to recover from the inferior initiation caused by this prescaling and consequently it leads to a test that has a simpler setting and requires less computation than the test in [7]. The organization of the paper follows that of the paper in [7]. Similarities between the two tests are used, when appropriate, to omit proofs and instead give reference to counterpart theorems proved in [7]. However, situations that involve differences are given independent full proofs.

The first tabular test for 2-D stability was proposed by Maria and Fahmy in [19]. It was based on an early version of a 1-D stability table by Jury. Anderson and Jury proposed to solve the problem by a polynomial Schur-Cohn matrix in [1]. To this end, Siljak showed in [21] that for testing 2-D stability via positive definiteness of the Schur-Cohn polynomial matrix over the unit circle, it suffices to test its definiteness at a single point and the positivity over $T$ of the determinant (that is, the positivity on $T$ of lower principal minor polynomials need not be examined). Jury designed a modified form of his 1-D stability test that produces explicitly the principal minors of the Schur-Cohn matrix [16], [17] that may be used for 2-D stability to combine the manageability of a tabular test with the computational saving offered by Siljak's simplification. More recently, Hu and Jury in [13] improved the test by removing from its implementation redundant factors. The Maria and Fahmy 2-D test and all subsequent 2-D tabular tests up until this improvement are of exponential order of complexity. To realize this, it suffices to observe that the degrees of the polynomials whose positivity over $T$ is to be tested grow exponentially with $n$ (say $n = n_1 = n_2$) [20]. The truth is (a fact that has been overlooked in past studies) that the computation of the "tables" used to obtain these polynomials generally requires many more operations than the testing of the last or even all the positivity conditions. The tests in [13] and those here will be shown to be of just a polynomial complexity of order $n^6$. The current test and the test in [7] may be regarded as the *immittance* counterpart to the *scattering* test in [13] (using terms suggested and motivated in [9] and other references therein). The classification is manifested by the properties that [13] involves a two-term recursion of 2-D polynomials or matrices with no particular structure, whereas here and in [7] the tests use a three-term recursion to propagate 2-D polynomials or matrices with a special symmetry (the matrices are centrosymmetric). A count of operations for the current test is also performed (and it also applies approximately to the test in [7]). The symmetry of the matrices in the immittance tabular tests gives them an edge over the scattering tabular test in [13], with regard to both cost of computation and numerical accuracy.

There exists in the literature on 2-D stability variations on the convention that we use in defining the problem in (1) and (2). Sometimes the polynomial in (1) is defined in negative powers of the variables and/or is regarded as stable if it does not vanish in $\bar{U} \times \bar{U}$. Adaptation to such alternative conventions may be reached by reversion of the coefficient matrix $D = (d_{i,k})$ [7].

The paper is organized as follows. Section 2 introduces the notation, cites a widely used simplification to condition (2), and derives a modification for the stability test in [3] that better suits the current application. Section 3 derives a preliminary form of the test, called the "F-table", and simplifies its initial large set of stability conditions. The F-table, can be shown to be of exponential complexity like the aforementioned older tabular tests. Section 4 reduces the row sizes of the F-table matrices by elimination of redundant polynomial factors and obtains simple stability conditions for the resulting reduced-size "E-table"—the final form of the table. Section 5 summarizes the new 2-D tabular stability test, carries an approximate count of operations, and illustrates the method by a simple numerical example. The paper ends with some concluding remarks.

## 2. Preliminaries

This section introduces the notation that will be used in this paper. It then provides a division-free version for the 1-D stability test in [6] that we shall use to test condition (2) in conjunction with the Huang-Strintzis simplification cited below in Lemma 1.

### 2.1. Notation

We shall use $P = (p_{i,k})$ to denote the coefficients matrix of a 2-D polynomial $P(s,z) = \sum_{i=0}^{n_1} \sum_{k=0}^{n_2} p_{i,k} s^i z^k$. Similarly, $p$ will denote the vector of coefficients of a 1-D polynomial $p(z)$. In correspondence to the polynomial variables $z$, $\mathbf{z}$ will denote a vector whose entries are powers in ascending degrees of the variable, $\mathbf{z} = [1, z, \ldots, z^i, \ldots]^t$ (of length determined by context). The notation admits reference to the above 2-D polynomial in several ways, including

$$P(s,z) = \sum_{k=0}^{n_2} p_k(s) z^k = [p_0(s), p_1(s), \ldots, p_{n_2}(s)]\mathbf{z} = \mathbf{s}^t P \mathbf{z}.$$

Here $p_k$ is the $(k+1)$th column of $P$ and $p_k(s) = \mathbf{s}^t p_k$ is the (polynomial) coefficient of $z^k$ when $P(s,z)$ is regarded as a 1-D polynomial in the variable $z$. This notation does not explicate the row indices of the entries of $P = (p_{i,k})$, which may be added as follows, $p_k = [p_{0,k}, p_{1,k}, \ldots, p_{n_1,k}]^t$, but mostly we shall manipulate vectors as a whole and act on columns of matrices. The superscript $\sharp$ will denote (conjugate-) reversion, defined for a matrix and a vector by

$$P^\sharp = J P^\star J, \qquad p^\sharp = J p^\star,$$

respectively, where $J$ denotes the reversion matrix with 1's on the main antidiagonal and zeros elsewhere, and $\star$ denotes complex conjugation.

Convolution will be denoted by $*$, e.g.,

$$h = f * p_k \longleftrightarrow h(s) = f(s)p_k(s).$$

Convolution of a vector by a matrix will mean column-by-column convolution, i.e.,

$$\begin{aligned} G = f * P = [f * p_0, f * p_1, \ldots, f * p_n] = [g_0, g_1, \ldots, g_n] \\ \longleftrightarrow G(s, z) = f(s)P(s, z) = [g_0(s), g_1(s), \ldots, g_n(s)]\mathbf{z} \end{aligned}$$

The converse operation of columnwise deconvolution (division with no remainder) will be denoted by

$$\begin{aligned} P = G \,/\, f = [p_0, p_1, \ldots, p_n] \\ \longleftrightarrow P(s, z) = \frac{G(s, z)}{f(s)} = [p_0(s), p_1(s), \ldots, p_n(s)]\mathbf{z} \end{aligned}$$

and it will represent extraction of a factor $f(s)$ common to all the polynomials $g_k(s)$. Notation like $[0, G]$ or $[G, 0, 0]$ will denote pre- or post-padding of $G$ by the depicted number of columns of zeros.

In the process of developing the new method (and only for this sake), it will be convenient to think of the coefficient matrix of $P(s, z)$ as associated with a "*balanced polynomial*," rather than $p_k(s)$. Accordingly, $P$ will be associated with the next alternative function,

$$P(\tilde{s}, z) = s^{-n_1/2} P(s, z) = \sum_{k=0}^{n_2} p_k(\tilde{s}) z^k = \tilde{\mathbf{s}}^t P\mathbf{z}, \tag{3}$$

where $\tilde{\mathbf{s}} := [s^{-m/2}, s^{-(m/2-1)}, \ldots, s^{(m/2-1)}, s^{m/2}]^t$, and $\tilde{s}$, as a function argument, denotes a power series to equal extent in the two variables $(s^{-1}, s)$ (in the current method, in contrast to [7], only the even integer $m$ will be encountered).

We shall construct for the polynomial $D(z_1, z_2)$ a sequence of centrosymmetric matrices (the "2-D table") $\{E_m\,,\ m = 0, 1, \ldots, n(= n_2)\}$, $E_m^\sharp = E_m$. These matrices may also be referred to as the polynomials $E_m(s, z)$ or $E_m(\tilde{s}, z)$. Each polynomial $E_m(s, z)$ will be of degree $n_2 - m$ in $z$ and of a certain degree in $s$ that increases exponentially as a function of $m$ in the initial form and only linearly in the final form of the sequence. When using the matricial notation, the sequential index $m$ will be set in brackets and precede other indices. For example, $E_m(\tilde{s}, z) = [e_{[m]\ 0}(\tilde{s}), e_{[m]\ 1}(\tilde{s}), \ldots, \ldots, e_{[m]\ n-m}(\tilde{s})]\mathbf{z} = \tilde{\mathbf{s}}^t [e_{[m]\ 0}, e_{[m]\ 1}, \ldots, e_{[m]\ n-m}]\mathbf{z}$, where $e_{[m]\ k} = [e_{[m]0,k}, e_{[m]1,k}, \ldots, e_{[m]\ell_e(m),k}]^t$ is the $(k + 1)$th column of $E_m$.

## *2.2. Huang-Strintzis stability conditions*

The most commonly used starting point for 2-D stability tests is the following lemma. It was introduced by Huang for $a = \infty$ [14], [20] and set by Strintzis [22] into its next more general form.

**Lemma 1.** *$D(z_1, z_2)$ is stable if and only if*

(a)

$$D(z, a) \neq 0 \quad \textit{for all} \quad z \in \bar{V} \textit{and some } a \in \bar{V} \tag{4}$$

(b)

$$D(s, z) \neq 0 \quad \textit{for all} \quad (s, z) \in T \times \bar{V} \ . \tag{5}$$

**Remark 1.** It is desirable to choose a real $a$ in order not to introduce unnecessary complex arithmetics when $D$ is real. We shall use this lemma with $a = 1$. This value integrates nicely with the special role that $z = 1$ plays in our underlying 1-D stability test.

### *2.3. Division-free 1-D stability test*

As is well known, stability of a 1-D discrete-time linear system of order $n$ corresponds to the requirement that its system polynomial $p(z)$ has its $n$ zeros in $U$ (is "stable"). This condition may be brought to closer terms with (2) by writing it as $p(z) \neq 0$ for all $z \in \bar{V}$. We intend to apply the test in [3] to (5), by regarding $D(s, z)$ as a 1-D polynomial in $z$ with coefficients dependent on $s$. However doing so in a direct manner would lead to the manipulation of a sequence of polynomials in $z$ with coefficients that are rational functions of $s$. In order to avoid rational functions, we derive here a division-free version for the 1-D stability test of [3].

The stability of a 1-D polynomial $p(z)$

$$p(z) = \sum_{k=0}^{n} p_k z^k, \qquad p(1) \neq 0 \tag{6}$$

may be tested using the following algorithm and Proposition 1.

**Algorithm 1. 1-D Table**.

Construct for the 1-D polynomial (6), the following sequence of symmetric polynomals $\{f_m(z) = \sum_{k=0}^{n-m} f_{[m]k} z^i,\ m = 0, 1, \ldots, n\}$, $f_m(z) = f_m^\sharp(z)$.

(i) **Initiation.** Form $\tilde{p}(z) = p(1)^\star p(z)$ and obtain

$$f_0(z) = \tilde{p}(z) + \tilde{p}^\sharp(z), \qquad f_1(z) = (\tilde{p}(z) - \tilde{p}^\sharp(z))/(z - 1) \tag{7}$$

(ii) **Recursion.** For $m = 1, \ldots, n - 1$:

$$z f_{m+1}(z) = (f_{[m-1]0} f_{[m]0}^\star + f_{[m-1]0}^\star f_{[m]0} z) f_m(z) - f_{[m]0} f_{[m]0}^\star f_{m-1}(z) \ . \tag{8}$$

**Proposition 1** (Stability Conditions for Algorithm 1)**.** *Assume that Algorithm 1 is applied to the polynomial (6). Then $p(z)$ is stable if and only if*

$$f_m(1) > 0 \qquad m = 1, \ldots, n. \tag{9}$$

**Proof.** The polynomials in Algorithm 1 and those used in [3] differ by scaling factors, say $f_m(z) = \psi_m t_m(z)$, where $t_m(z) = \sum_{k=0}^{n-m} t_{[m]k} z^i$. (Note that the polynomials in [3] $T_m(z)$ were indexed in reversed order, $t_m(z) = T_{n-m}(z)$.) A comparison of the recursions in Algorithm 1 with the recursion in [3] reveals the relations $\psi_0 = 1$, $\psi_1 = 1$, $\psi_m = |f_{[m-1]0}|^2 \psi_{m-2} = |t_{[m-1]0}|^2 \psi_{m-1}^2 \psi_{m-2}$, $m \geq 2$. The current necessary and sufficient conditions follow from corresponding stability conditions in [3] via the fact that all the $\psi_m$ are real and positive. □

**Remark 2.** The conditions $f_{[m]0} \neq 0$ for all $m$ are necessary conditions for stability and they correspond to "normal conditions" in [3]. However, currently they need no special care because the algorithm avoids the division operation. The instability that a vanishing $f_{[m]0}$ implies is detected as a subsequent violation of (9). All the polynomials $f_m(z)$ are symmetric, $f_{m,n-m-i} = f_{m,i}^\star$, $i = 0, \ldots, n-m$. This fact implies in particular that all $f_m(1)$ are real, so that writing $f_m(1) > 0$ makes sense.

## 3. An intermediate test form

In order to apply the above 1-D stability test to $D(z_1, z_2)$, we define

$$M(\tilde{s}, z) := D(s^{-1}, 1) D(s, z) = \tilde{\mathbf{s}}^t M \mathbf{z}. \tag{10}$$

Note that condition (b) of Lemma 1 holds if and only if $M(\tilde{s}, z) \neq 0 \, \forall (s, z) \in T \times \bar{V}$. So, the division-free 1-D stability test is applicable to $p_s(z) = D(s, z)$ regarding it as a polynomial in $z$ with coefficients dependent on $s \in T$. It involves Algorithm 1, using $\tilde{p}_s(z) = M(\tilde{s}, z)$ in (7), and Proposition 1. Also note that meeting the requirement $D(s, 1) \neq 0$ implies that $\tilde{p}_s(1)$ is real and positive ($= |D(\tilde{s}, 1)|^2 > 0$). Also we draw attention to the fact that for balanced polynomial coefficients, complex conjugation for values of $s \in T$ amounts to reversion of the column vectors. The important advantage of using the balanced polynomial view is that complex conjugation in Algorithm 1 is implemented in the following Algorithm 2 by reversion of rows without increasing the row sizes of the coefficient matrices.

### 3.1. *Table construction*

The first form for the 2-D stability table (which will serve as a basis for further improvements) follows from the application of Algorithm 1 to $M(\tilde{s}, z)$. Here and through most of the derivation we shall use polynomial rather than matricial notation. However, the introduced notation convention allows a simple translation of the polynomial algorithm to a matricial form. At the end of the derivation, in Section 5, the method will be summarized using matricial notation.

**Algorithm 2. The F-table** (an intermediate 2-D table form). Construct for the tested $D(z_1, z_2)$ a sequence of polynomials $\{F_m(\tilde{s}, z) = \sum_{k=0}^{n-m} f_{[m]\,k}(\tilde{s})z^k,\ m = 0, 1, \ldots, n\ (= n_2)\}$ by the following recursion.

(i) **Initiation.**

$$M(\tilde{s}, z) := D(s^{-1}, 1)D(s, z)$$

$$F_0(\tilde{s}, z) = M(\tilde{s}, z) + M^{\sharp}(\tilde{s}, z), \qquad F_1(\tilde{s}, z) = \frac{M(\tilde{s}, z) - M^{\sharp}(\tilde{s}, z)}{z - 1} \tag{11}$$

(ii) **Recursion.** For $m = 1, 2, \ldots, n-1$, obtain $F_{m+1}(\tilde{s}, z)$ by

$$h_m(\tilde{s}) = f_{[m-1]\,0}(\tilde{s}) f^{\sharp}_{[m]\,0}(\tilde{s})$$
$$r_m(\tilde{s}) = f_{[m]\,0}(\tilde{s}) f^{\sharp}_{[m]\,0}(\tilde{s})$$

$$zF_{m+1}(\tilde{s}, z) = h_m(\tilde{s})F_m(\tilde{s}, z) + h^{\sharp}_m(\tilde{s})\, z\, F_m(\tilde{s}, z) - r_m(\tilde{s})F_{m-1}(\tilde{s}, z). \tag{12}$$

It follows from the symmetry of the polynomials in Algorithm 1 that the 2-D polynomials $F_m(s, z)$ are centrosymmetric, by which we mean that their coefficient matrices satisfy the symmetry $F_m = F^{\sharp}_m$.

### 3.2. *Stability conditions*

Define for Algorithm 2 the sequence

$$\varphi_m(\tilde{s}) := F_m(\tilde{s}, 1) = \sum_{k=0}^{n-m} f_{[m]\,k}(\tilde{s}), \qquad m = 0, \ldots, n. \tag{13}$$

$\varphi_m(\tilde{s})$ are (balanced) symmetric polynomials, $\varphi^{\sharp}_m = \varphi_m$. This implies in particular that $\varphi_m(\tilde{s})$ is real $\forall s \in T$.

**Proposition 2** (Stability Conditions for F-Table)**.** *$D(z_1, z_2)$ is stable if and only if the following conditions (a) and (b) or (b') hold.*

(a) *$D(z, 1) \neq 0$ for all $z \in \bar{V}$,*
(b) *$\varphi_m(\tilde{s}) > 0$, $m = 1, \ldots, n$ for all $s \in T$,*
(b') *(i) $D(1, z) \neq 0$ for all $z \in \bar{V}$,*
*(ii) $\varphi_m(s) \neq 0, m = 1, \ldots, n$ for all $s \in T$,*

*where $\varphi_m(\tilde{s})$ are defined in (13) for the F-table of $D(z_1, z_2)$.*

**Proof.** We need to show that conditions (b) here and in Lemma 1 are equivalent. If $D(z_1, z_2)$ is 2-D stable, then (a) holds, and Algorithm 2, regarded, for each $s \in T$, as a 1-D test by Algorithm 1 for $\tilde{p}_s(z) := D^{\sharp}(\tilde{s}, 1)D(\tilde{s}, z)$, implies the conditions in (b), by Proposition 1. Conversely, if (a) and (b) hold then, reversing the argument, (a) admits the application of Algorithm 1 to $\tilde{p}_s(z)$, and

Proposition 1 implies the $D(s, z) \neq 0\ \forall z \in \bar{V}$. Therefore, $D(z_1, z_2)$ is (2-D) stable by Lemma 1.

It remains to show that (b) is replaceable by (b'). Clearly, condition (b) is equivalent to the following two conditions: (1) $\varphi_m(\tilde{s}) \neq 0$, $m = 1, \ldots, n$ plus positivity at one point on $T$, say (2) $\varphi_m(1) > 0$, $m = 1, \ldots, n$. However, (2) is by Proposition 1 equivalent to 1-D stability of $M(1, z) = D(1, 1)D(1, z)$ and is well presented by (i) of (b'). Also (1) is equivalent to (ii) of (b') because for $s \in T$, $\varphi_m(\tilde{s}) \neq 0$ if and only if $\varphi_m(s) \neq 0$. □

**Remark 3.** A difference between the test here and the one in [7] is notable from this proposition and on. The stability conditions here are posed on $F_m(\tilde{s}, 1)$ directly. In contrast, in [7] they are posed on $F_m(\tilde{s}, 1)/F_0(\tilde{s}, 1)$. Similar differences will exist between all subsequent stability conditions here and their counterparts in [7]. The above proof is simpler than the proof of the corresponding property in [7] (because there additional care was needed to account for $F_0(s, 1)$, which may have there zeros on $T$ without implying instability).

**Remark 4.** Condition (b') is preferable over (b). The 1-D stability test for $D(1, z)$ may be carried out before starting the construction of the table. This way, in a case when $D(1, z)$ is found not to be stable, the table construction may be avoided. An equal computational effort is required to test algebraically the condition $\varphi_m(s)$ (not real valued on $T$) for no zeros on $T$ and the condition $\varphi_m(\tilde{s}) > 0\ \forall s \in T$. The stability of $D(1, z)$ will be stated as part of all subsequent stability conditions with both $\varphi_m(s) \neq 0$ or $\varphi_m(\tilde{s}) > 0\ \forall s \in T$ (and they both will be referred to as *positivity* conditions).

The sequence $\{\varphi_m(\tilde{s})\}$ obeys the next recursive relation, which will be instrumental in subsequent proofs:

$$\varphi_{m+1}(\tilde{s}) = h_m^r(\tilde{s})\varphi_m(\tilde{s}) - r_m(\tilde{s})\varphi_{m-1}(\tilde{s}), \quad \text{where} \quad h_m^r(\tilde{s}) := h_m(\tilde{s}) + h_m^\sharp(\tilde{s})\,. \tag{14}$$

It is obtained by setting $z = 1$ in Algorithm 2. Note that this recursion is not enough by itself to derive the sequence $\{\varphi_m(\tilde{s})\}$ because it requires $\{F_m(\tilde{s}, z)\}$ for $h_m^r(\tilde{s})$ and $r_m(\tilde{s})$. The next corollary states properties of $h_m^r(\tilde{s})$ and $r_m(\tilde{s})$ that will be needed in forthcoming proofs. They can be shown using the definition of these functions, the above recursion, and Proposition 2 (the proof is detailed in [7]).

**Corollary from Proposition 2.** *If $D(z_1, z_2)$ is stable, then*

(i) $h_m^r(\tilde{s}) > 0$ *for all $s \in T$ and all $m = 0, \ldots, n-1$.*
(ii) $r_m(\tilde{s}) > 0$ *for all $s \in T$ and all $m = 0, \ldots, n-1$.*

### *3.3. Refined stability conditions*

The next proposition simplifies Proposition 2 by showing that it suffices to carry out only the last positivity test. This is a simplification of the kind introduced to the 2-D stability literature by Siljak [21].

**Proposition 3** (Refined Stability Conditions for F-Table). *$D(z_1, z_2)$ is stable if and only if the three following conditions (i), (ii), and (iii) or (iii′) hold:*

(i) $D(z, 1) \neq 0$ *for all* $z \in \bar{V}$
(ii) $D(1, z) \neq 0$ *for all* $z \in \bar{V}$
(iii) $\varphi_n(s) \neq 0$ *for all* $s \in T$
(iii′) $\varphi_n(\tilde{s}) > 0$ *for all* $s \in T$

**Proof.** These are evidently necessary conditions because they form a subset of the conditions in Proposition 2. Sufficiency can be proved by using the recursion for $\varphi_m(\tilde{s})$ (14) and proof by contradiction. Assume that conditions (i)-(iii) hold and that nevertheless there exist $\varphi_m(\tilde{s})$ $m < n$ that vanish for $s \in T$. Proposition 1 implies that for all $m$, $\varphi_m(1) > 0$. Let $k$ be the least $m$ for which $\varphi_m(\tilde{s}) = 0$ and let $s_1 \in T$ be the closest point to $s = 1$ such that $\varphi_k(\tilde{s}_1) = 0$. Then by (14), $\varphi_{k+1}(\tilde{s}_1) = -r_k(\tilde{s}_1)\varphi_{k-1}(\tilde{s}_1) \leq 0$ because $r_k(\tilde{s}) \geq 0$ $\forall s \in T$ (by definition) and $\varphi_{k-1}(\tilde{s}_1) > 0$ (by the assumed choice of $k$). Therefore, $\varphi_{k+1}(\tilde{s})$ must vanish for some $s \in T$ closer to $s = 1$ than $s_1$. Let $s_2$ be the root of $\varphi_{k+1}(\tilde{s}) = 0$ on $T$ closest to $s = 1$, i.e., with maximal $s_2^R$, $s_1^R \leq s_2^R < 1$, where $s_i^R = \mathcal{R}e\{s_i\}$. Repeating this reasoning enough times implies that $\varphi_n(\tilde{s})$ must vanish for some $s \in T$ whose real part is in a subinterval $[s_{n_o}^R, 1)$ such that $s_{n_o}^R < 1$. This is a contradiction to the assumption that condition (iii) holds. Therefore, conditions (i)–(iii) imply the larger set of conditions of Proposition 2, and therefore are sufficient for stability. □

## 4. The 2-D stability test (final form)

In this section we first show that the F-table is of size higher than necessary. We then obtain a modified recursion that produces matrices of lower row size and finally derive stability conditions for the reduced-size table.

### *4.1. Redundant factors*

The next two lemmas reveal that each $F_m(\tilde{s}, z)$ $m \geq 2$ contains separable polynomial in $\tilde{s}$ factors (i.e., factors that divide $F_m(\tilde{s}, z)$ with no remainder) that are passed to all subsequent $F_{m+i}(\tilde{s}, z)$, $i > 0$, and causes a rapid growth of the degrees in $s$ of these polynomials.

**Lemma 2.** *$\varphi_0(\tilde{s})$ divides $F_2(\tilde{s}, z)$.*

**Proof.** Note that $\varphi_0(\tilde{s}) = F_0(\tilde{s}, 1) = 2D(s, 1)D(s^{-1}, 1)$. Using the notation

$$D(\tilde{s}, z) = \sum_{k}^{n} d_k(\tilde{s}) z^k, \qquad \delta(\tilde{s}) = \sum_{k=0}^{n} d_k(\tilde{s}) \ ,$$

$M(\tilde{s}, z) = \delta(\tilde{s})^{\sharp} D(\tilde{s}, z)$ and $\varphi_0(\tilde{s}) = 2\delta(\tilde{s})\delta^{\sharp}(\tilde{s})$. It is easy to see that

$$f_{[0]\,0}(\tilde{s}) = \delta(\tilde{s})d_n^{\sharp}(\tilde{s}) + \delta^{\sharp}(\tilde{s})d_0(\tilde{s})$$

$$f_{[1]\,0}(\tilde{s}) = \delta(\tilde{s})d_n^{\sharp}(\tilde{s}) - \delta^{\sharp}(\tilde{s})d_0(\tilde{s}).$$

In the following discussion, we shall use the notation $\mapsto$ to mean that the right-hand side is what remains after evaluating the left-hand side and dropping, from a sum of terms, terms already seen to contain the factor $\delta(\tilde{s})\delta^{\sharp}(s)$. Evaluate $h_1(\tilde{s})$ and $r_1(\tilde{s})$,

$$h_1(\tilde{s}) = f_{[0]\,0}(\tilde{s})f^{\sharp}_{[1]\,0}(\tilde{s}) \mapsto -\delta(\tilde{s})^2 d_0^{\sharp}(\tilde{s})d_n^{\sharp}(\tilde{s}) + \delta^{\sharp}(\tilde{s})^2 d_0(\tilde{s})d_n(\tilde{s}) =: \hat{h}_1(\tilde{s})$$

$$r_1(\tilde{s}) = f_{[1]\,0}(\tilde{s})f^{\sharp}_{[1]\,0}(\tilde{s}) \mapsto -\delta(\tilde{s})^2 d_0^{\sharp}(\tilde{s})d_n^{\sharp}(\tilde{s}) - \delta^{\sharp}(\tilde{s})^2 d_0(\tilde{s})d_n(\tilde{s}) =: \hat{r}_1(\tilde{s}).$$

Therefore,

$$zF_2(\tilde{s}, z) \mapsto (\hat{h}_1(\tilde{s}) + z\hat{h}_1^{\sharp}(\tilde{s}))F_1(\tilde{s}, z) - \hat{r}_1(\tilde{s})F_0(\tilde{s}, z) =$$

$$-\hat{h}_1(\tilde{s})(\delta^{\sharp}(\tilde{s})D(\tilde{s}, z) \; - \; \delta(\tilde{s})D^{\sharp}(\tilde{s}, z)) - \hat{r}_1(\tilde{s})(\delta^{\sharp}(\tilde{s})D(\tilde{s}, z) + \delta(\tilde{s})D^{\sharp}(\tilde{s}, z)),$$

where the $\mapsto$ convention allows the substitutions $h_1(\tilde{s}) \mapsto \hat{h}_1(\tilde{s})$ and $r_1(\tilde{s}) \mapsto \hat{r}_1(\tilde{s})$. The second equality follows from the definition of $F_0(\tilde{s}, z)$ and $F_1(\tilde{s}, z)$ and from using $\hat{h}_1^{\sharp}(\tilde{s}) = -\hat{h}_1(\tilde{s})$ to cancel out the $(z - 1)$ in the denominator of the definition of $F_1(\tilde{s}, z)$. Now we insert the expressions for $\hat{h}_1(\tilde{s})$ and $\hat{r}_1(\tilde{s})$ and continue the evaluation process using $\mapsto$ to drop factors of $\delta(\tilde{s})\delta^{\sharp}(\tilde{s})$ from the sum of terms,

$$\begin{aligned} &\mapsto -\delta(\tilde{s})^3 d_0^{\sharp}(\tilde{s})d_n^{\sharp}(\tilde{s})D^{\sharp}(\tilde{s}, z) - \delta(\tilde{s})^{\sharp^3} d_0(\tilde{s})d_n(\tilde{s})D(\tilde{s}, z) \\ &\quad +\delta(\tilde{s})^3 d_0^{\sharp}(\tilde{s})d_n^{\sharp}(\tilde{s})D^{\sharp}(\tilde{s}, z) + \delta(\tilde{s})^{\sharp^3} d_0(\tilde{s})d_n(\tilde{s})D(\tilde{s}, z) = 0\,. \end{aligned}$$

We have therefore obtained that $F_2(\tilde{s}, z)$ is composed of a sum of terms that contain the factor $\delta(\tilde{s})\delta(\tilde{s})^{\sharp}$ plus terms that sum up to zero. This completes the proof. $\square$

**Lemma 3.** *Consider a sequence $\{F_m(\tilde{s}, z)\}_0^n$ produced by the recursion (12).*

(a) *If $f(\tilde{s})$ is a factor of $F_m(\tilde{s}, z)$ $m \geq 0$, then it is a factor of all subsequent $F_{m+i}(\tilde{s}, z)$, $i \geq 1$.*
(b) *For any four consecutive polynomials $G_0(\tilde{s}, z)$, $G_1(\tilde{s}, z)$, $G_2(\tilde{s}, z)$, $G_3(\tilde{s}, z)$ in this sequence, $g_{[1]\,0}(\tilde{s})g^{\sharp}_{[1]\,0}(\tilde{s})$ is a factor of $G_3(\tilde{s}, z) = \sum g_{[3]\,i}(\tilde{s})z^i$. Namely, $g_{[1]\,0}(\tilde{s})g^{\sharp}_{[1]\,0}(\tilde{s})$ divides exactly (with no remainder) each $g_{[3]\,i}(\tilde{s})$.*

The assertion in Lemma 3 is a property of the recursion (12). The requested proof has already been given in [7] because the two tests use the same recursion form (12). It follows from property (b) that any $F_{k+2}(\tilde{s}, z)$, $k = 0, 1, \ldots,$ is divisible by each of the factors $f_{[i]\,0}(\tilde{s})f^{\sharp}_{[i]\,0}(\tilde{s})$, $i = 0, \ldots, k$. By property (a), such factors build up as the recursion goes on and their multiplicity increases.

The factor in Lemma 2 was not observed in [4]. It is of a different nature than the factors in Lemma 3. It stems from the multiplication by $D(s^{-1}, 1)$ of $D(s, z)$ in forming $M(\tilde{s}, z)$. The factor $\varphi_0(\tilde{s})$ is by definition nonnegative on $T$, and for a stable $D(z, 1)$, it is strictly positive, $\varphi_0(\tilde{s}) > 0\ \forall s \in T$. By property (a) of Lemma 3, the $2n_1$ degree polynomial $\varphi_0(s)$ is a factor of all $F_m(s, z)$, $m \geq 2$. It is important to realize that its elimination from $F_2(\tilde{s}, z)$ does not interfere with the mechanism that builds up the factors figured in Lemma 3.

### 4.2. *Table construction*

The next algorithm shows how to produce efficiently a sequence of matrices $\{E_m(\tilde{s}, z),\ m = 0, \ldots, n\}$ in which each $E_m(\tilde{s}, z)$ presents the result of removing from $F_m(\tilde{s}, z)$ all common factors that it accumulates according to Lemmas 2 and 3. The efficiency is manifested in obtaining the $E_m$'s directly rather than the more obvious alternative of removing the common factors after completing the F-table. Specifically in the following algorithm, the division by $q_0$ takes care of the factor revealed in Lemma 2, and subsequent $q_m(\tilde{s})$, $m \geq 1$, remove the factors of Lemma 3 as soon as possible.

**Algorithm 3. E-table (final form).**
Construct for $D(z_1, z_2)$ a sequence of polynomials $\{E_m(\tilde{s}, z) = \sum_{k=0}^{n-m} e_{[m]\,k}(\tilde{s})z^k,\ m = 0, 1, \ldots, n(= n_2)\}$, where $E_m = E_m^\sharp\ \forall m$, as follows.

(i) **Initiation.** $M(\tilde{s}, z) = D(s^{-1}, 1)D(s, z)$

$$E_0(\tilde{s}, z) = M(\tilde{s}, z) + M^\sharp(\tilde{s}, z), \qquad E_1(\tilde{s}, z) = \frac{M(\tilde{s}, z) - M^\sharp(\tilde{s}, z)}{z - 1} \tag{15}$$

$$q_0(\tilde{s}) = E_0(\tilde{s}, 1)$$

(ii) **Recursion.** For $m = 1, \ldots, n - 1$, obtain $E_{m-1}(\tilde{s}, z)$ by

$$g_m(\tilde{s}) = e_{[m-1]\,0}(\tilde{s})e^\sharp_{[m]\,0}(\tilde{s})$$

$$q_m(\tilde{s}) = e_{[m]\,0}(\tilde{s})e^\sharp_{[m]\,0}(\tilde{s})$$

$$zE_{m+1}(\tilde{s}, z) = \frac{g_m(\tilde{s})E_m(\tilde{s}, z) + g_m^\sharp(\tilde{s})zE_m(\tilde{s}, z) - q_m(\tilde{s})E_{m-1}(\tilde{s}, z)}{q_{m-1}(\tilde{s})}. \tag{16}$$

### 4.3. *Stability conditions*

For the derivation of stability conditions for the E-table, we shall need the relation between the sequences $\{F_m(\tilde{s}, z)\}$ and $\{E_m(\tilde{s}, z)\}$. They evidently are of the form

$$F_m(\tilde{s}, z) = \alpha_m(\tilde{s})E_m(\tilde{s}, z), \tag{17}$$

where $\alpha_m(s)$ are symmetric, $\alpha_m^\sharp(s) = \alpha_m(s)$. The degree of $\alpha_m(s)$ represents the amount of row reduction achieved by replacing $F_m$ by $E_m$. Substituting the above relation into one three-term recursion and then comparing it to the other gives the next set of recursive rules for $\alpha_m(s)$. $\alpha_0(\tilde{s}) = \alpha_1(\tilde{s}) = 1$, and

$$\alpha_{m+1}(\tilde{s}) = \alpha_m^2(\tilde{s})\alpha_{m-1}(\tilde{s})q_{m-1}(\tilde{s}), \qquad m \geq 1, \tag{18}$$

or

$$\alpha_{m+1}(\tilde{s}) = \frac{\alpha_m^2(\tilde{s})}{\alpha_{m-1}(\tilde{s})} r_{m-1}(\tilde{s}), \qquad m \geq 1. \tag{19}$$

In correspondence to the definition of $\varphi_m(\tilde{s})$ for the F-table, we define for the E-table the (balanced) symmetric polynomial sequences $\{\epsilon_m(\tilde{s})\}$ by

$$\epsilon_m(\tilde{s}) := E_m(\tilde{s}, 1), \qquad m = 0, \ldots, n \tag{20}$$

Setting $z = 1$ in Algorithm 3 shows that $\epsilon_m(\tilde{s})$ obey the recursion: $\epsilon_0(\tilde{s}) = E_0(\tilde{s}, 1) = 2D(\tilde{s}, 1)D^\sharp(\tilde{s}, 1)$, $\epsilon_1(\tilde{s}) = E_1(\tilde{s}, 1)$,

$$\epsilon_{m+1}(\tilde{s}) = \frac{g_m^r(\tilde{s})\epsilon_m(\tilde{s}) - q_m(\tilde{s})\epsilon_{m-1}(\tilde{s})}{q_{m-1}(\tilde{s})}, \qquad m = 1, \ldots, n-1, \tag{21}$$

where we define $g_m^r(\tilde{s}) := g_m(\tilde{s}) + g_m^\sharp(\tilde{s})$.

The next stability conditions are the E-table counterpart of Proposition 1.

**Proposition 4** Stability condition for the E-table**.** *$D(z_1, z_2)$ is stable if and only if the three conditions (i), (ii), and (iii) or (iii′) hold.*

(i) $D(z, 1) \neq 0$ *for all* $z \in \bar{V}$.
(ii) $D(1, z) \neq 0$ *for all* $z \in \bar{V}$.
(iii) $\epsilon_m(\tilde{s}) \neq 0$, $m = 1, 2, \ldots, n$, *for all* $s \in T$
(iii′) $\epsilon_m(s) \neq 0$, $m = 1, \ldots, n$, *for all* $s \in T$,

*where $\{\epsilon_m(\tilde{s})\}$ are defined in (20) for the E-table of $D(z_1, z_2)$.*

**Proof.** If $D(z_1, z_2)$ is stable, then by the corollary from Proposition 2, all $\varphi_m(\tilde{s})$ and $r_m(\tilde{s})$ are positive on $T$. Therefore, all $\alpha_m(\tilde{s}) > 0$ on $T$ by (19). Therefore, the conditions follow from Proposition 2. via (17).

Assume that the three conditions hold. We have by definition that $q_m(\tilde{s}) \geq 0$ on $T$ and want to show that they are strictly positive there. Assume the converse, that $q_m(\tilde{s}_o) = 0$ for some $m$ and $\tilde{s}_o \in T$. Then $e_{[m]\,0}(s_o) = 0$ and in turn $g_m(s_o) = 0$. The latter leads via (21) to the contradiction $\epsilon_{m+1}(\tilde{s_o}) = 0$. Therefore, $q_m(\tilde{s}) > 0$ on $T$ for $m = 0, \ldots, n-1$. It follows via (18) that all $\alpha_m(\tilde{s}) > 0$ on $T$. Therefore, all $\varphi_m(\tilde{s}) > 0$ on $T$ by (17). Consequently, the conditions here imply the sufficient conditions for stability in Proposition 2. □

The main theorem shows that the single positivity condition of Proposition 3 for the F-table also remains valid for the reduced size E-table.

**Main Theorem.** *$D(z_1, z_2)$ is stable if and only if the following three conditions (i), (ii), and (iii) or (iii′) hold:*

(i) $D(z, 1) \neq 0$ *for all* $z \in \bar{V}$,
(ii) $D(1, z) \neq 0$ *for all* $z \in \bar{V}$,
(iii) $\epsilon_n(s) \neq 0$ *for all* $s \in T$,
(iii′) $\epsilon_n(s) > 0$ *for all* $s \in T$,

*where* $\epsilon_n$ *is defined in (20) for the E-table of* $D(z_1, z_2)$.

**Proof.** The stated conditions form a subset of the necessary conditions in Proposition 4 and are therefore necessary for stability.

Sufficiency follows by showing that the proof for Proposition 3 is extendable to the E-table. It is possible to repeat the argument there and show that any assumed $\epsilon_k(\tilde{s}_1) = 0$ for $k < n$ would imply the contradiction that $\epsilon_n(\tilde{s}_o) = 0$ for some $s_o \in T$ by using this time recursion (21). The same proof still works because a zero on $T$ of $\epsilon_k(\tilde{s})$ immediately affects the next $\epsilon_{k+1}(\tilde{s})$. In difference, a $q_m(\tilde{s})$ factor that is formed at step $k$ divides the right-hand side of (16) only at step $k+2$. Consequently, a zero on $T$ of $\epsilon_k(\tilde{s})$ will pass to subsequently indexed $\epsilon_m(\tilde{s})$ before division by $q_m(\tilde{s})$ may possibly cancel it out. So once again, any $\epsilon_m(\tilde{s}_o) = 0$ for $s_o \in T$ and $m < n$ contradicts condition (iii) of this theorem. Therefore, the three conditions here imply the three conditions in Proposition 4, which imply that $D(z_1, z_2)$ is stable. □

The main theorem shows that the simplification of replacing the F-table by the E-table does not complicate the simplicity of the associated stability conditions. As a matter of fact, the single positivity test of $\epsilon_n(s)$ is simpler than testing $\varphi_n(s)$ because $\epsilon_n(s)$ has a much lower degree than $\varphi_n(s)$. The complexity reduction in the construction of the E-table is even more impressive. Let $\ell_f(m)$ and $\ell_e(m)$ denote the degree of $F_m(s, z)$ and $E_m(s, z)$ in $s$, respectively. $\ell_f(m)$ is seen from the recursion form (12) to satisfy the difference equation

$$\ell_f(m+2) - 2\ell_f(m+1) - \ell_f(m) = 0. \tag{22}$$

A closed-form expression for $\ell_f(m)$ can be obtained by solving it for the initial conditions $\ell_f(-1) = \ell_f(0) = n_1$. The solution is a linear combination of $\lambda_1^m$ and $\lambda_2^m$ with $\lambda_{1,2} = 1 \pm \sqrt{2}$. Therefore, the solution increases exponentially with $m$. In difference, $\ell_e(m)$ is seen from recursion (16) to obey the equation

$$\ell_e(m+2) - 2\ell_e(m+1) + \ell_e(m) = 0 \tag{23}$$

whose solution for the initial values $\ell_e(1) = 2n_1$ and $\ell_e(2) = 4n_1$ is $\ell_e(m) = 2mn_1$. Therefore, the number of rows in the matrices of the E-table increases only linearly with $m$. We skip the count of operations for the F-table but will carry out in the next section an approximate count for the E-table. Note that the degree of $\epsilon_n(s) = E_n(s, z)$, the last and only polynomial whose positivity needs to be tested, is $2n_1n_2$.

It is interesting to look into the effect of missing the factor depicted in Lemma 2 in [4]. Not dividing $E_2(\tilde{s}, z)$ by $q_0(\tilde{s})$ leaves the algorithm and stability conditions equally valid, but the following changes occur. The degree of $E_2(s, z)$ in

$s$ becomes $\ell_e(2) = 6n_1$ (instead of $4n_1$). The solution of (23) for $\ell_e(1) = 2n_1$, $\ell_e(2) = 6n_1$ becomes $\ell_e(m) = (4m-2)n_1$. This would imply the (almost) doubling of the row sizes of subsequent matrices and an almost factor of 4 increase in the amount of computation required for the construction of the table (see the count in Section 5.2). The final positivity test would involve a symmetric polynomial of degree $2n_1(2n_2-1)$ instead of just $2n_1n_2$.

## 5. Summary and illustration

The proposed 2-D stability test is summarized in this section. The next summary uses the alternative matrix notation shown in Section 2.1. This way of presentation enlightens the simple implementation of this tabular test by an array-oriented language. Also in this section, we evaluate the computational cost of the method and illustrate it by a numerical example.

### *5.1. The proposed 2-D stability tabular test*

In the next presentation of the procedure, we add in brackets optional steps (steps that are suggested but may be skipped). They present a collection of necessary conditions for 2-D stability beyond the minimal set of necessary and sufficient conditions that were asserted in the route to the main theorem. If observed, they may help to detect earlier instability and consequently save the remaining computation.

To test whether $D(z_1, z_2)$ (1) is stable, i.e., (2) holds, proceed as follows. (In the following, 'exit' marks points at which the conclusion that $D(z_1, z_2)$ is not stable is reached and therefore the procedure may be terminated.)

**1. Preliminary 1-D tests:**

*Test whether $D(z, 1)$ is 1-D stable.*

False - 'exit', True - continue.

*Test whether $D(1, z)$ is 1-D stable.*

False - 'exit', True - continue.

[Optionally, perform additional tests for 1-D polynomials whose stability are necessary for 2-D stability, e.g., $D(s, z)$ and $D(z, s)$ at $s = \infty, -1$ or $D(z, z)$ and 'exit' if any of them is not stable.]

**2. Table Construction:** Form ($n = n_2$)

$$\delta = \sum_{k=0}^{n} d_k, \quad M = \delta^\sharp * D$$

$$E_0 = M + M^\sharp, \qquad E_1 = (M - M^\sharp)/[-1, 1]^t$$

$$q_0 = \sum_{k=0}^{n} e_{[0]\,k}.$$

For $m = 1, \ldots, n-1$ do:
[Optional (for $m \geq 1$): compute $\epsilon_m = \sum_{k=0}^{n-m} e_{[m]\,k}$,
*Test whether* $\mathbf{s}^t \epsilon_m \neq 0 \; \forall s \in T$
False - 'exit' , True - continue.]

$$q_m = e_{[m]\,0} * e^{\sharp}_{[m]\,0};\; g_m = e_{[m-1]\,0} * e^{\sharp}_{[m]\,0};\; \tilde{E}_m := g_m * [E_m, 0]$$

$$[0, E_{m+1}, 0] = \left(\tilde{E}_m + \tilde{E}^{\sharp}_m - q_m * E_{m-1}\right) / q_{m-1}$$

**3. Positivity Test:**
*Test whether* $\mathbf{s}^t e_{[n]\,0} \neq 0 \; \forall s \in T$
False - 'exit' , True - $D(z_1, z_2)$ is stable.

As is well known, convolution may be regarded as the algebraic operation of premultiplying a vector by a lower triangular Toeplitz matrix defined by the other vector. Similar implementation of the deconvolution is possible using instead the (simple-to-calculate) inverse of that matrix. The centrosymmetry of the matrices should be used to compute only half of their entries. The best way to exploit this symmetry is to calculate the upper half of the rows (and *not* the half of the columns). Each convolution should be carried out until only half of its full length. As is explained in the next subsection, this manner benefits both the amount and the accuracy of the computation more than the alternative approach of computing half of the columns.

### 5.2. *Computational cost*

A count of operations for the construction of the table involves the following observations. The symmetry of the matrices $E_m$ allows the computation of only half of the entries of each matrix. Using this symmetry to compute the upper half of the rows requires less computation and is numerically more robust than using the symmetry to compute the left half of the columns. The reason is that convolution of all columns (multiplication of two polynomials in $s$) halfway (until half of the full degree of the product polynomial) requires less computation and accumulates less numerical error then convolution of half of the columns to full extent. The degree of $E_m(s, z)$ is $(\ell(m), n2 - m)$, where $\ell(m) = 2mn_1$ for $m \geq 1$ $(\ell(0) = 2mn_1)$. The $m$th step of the recursion consists of three convolutions per column; $g_m(s)$ of degree $(4m-2)n_1$ multiplies $e_{[m]\,k}(s)$ of degree $2mn_1$, $q_m(s)$ of degree $4mn_1$ multiplies $e_{[m-1]\,k}(s)$ of degree $(2m-2)n_1$, the resulting numerator polynomial (of degrees $(6m-2)n_1$) is finally divided by $q_{m-1}(s)$ of degree $(2m-2)n_1$. Multiplication of two 1-D polynomials of degrees $k_1$ and $k_2$ requires $(k_1+1)(k_2+1)$ arithmetic operations. (The same count is also required for dividing them when one is a factor of the other.) Carrying the multiplication to only half degree of the product polynomial (as our case admits) requires only $1/4$ of the mentioned count. Avoiding the more lengthy details of a full exact count of operations, it is easy to conclude from these guidelines that the 2-D table's construction requires approximately $\frac{5}{6}n_1^2n_2^4 + O(n_{1,2}^5)$ flops, where $+O(n_{1,2}^5)$ is used to denote that other additive terms with powers $n_1^{\alpha_1}n_2^{\alpha_2}$ such that $\alpha_1 + \alpha_2 \leq 5$ are neglected. The test also involves two 1-D tests of degree $n_1$ and $n_2$ in step 1 and a unit circle test in step 3 for a polynomial of degree $2n_1n_2$. Using the 1-D stability test that underlies this 2-D tabular test, i.e., the test in [2], [3], the zero location of a polynomial of degree $n$ with respect to the unit circle requires $0.25n^2 + O(n)$ multiplications. Thus the cost requirement of the three 1-D tests culminates in step 3, which requires $n_1^2n_2^2 + O(n_{1,2}^3)$. This count is negligible compared to the complexity of the table's completion. (As a matter of fact, carrying out all $n$ positivity tests of Proposition 4 still involves one order of magnitude less computation than the cost of the table's construction.)

The above count of operations also approximates the other immittance tabular test in [7]. However, the two tests create different sequences of matrices (or 2-D polynomials) for the tested polynomial and differ in details. The test in [7] is based on the modified 1-D stability test in [6] and assigns to the tested 2-D polynomial a sequence that begins with an $E_{-1}(\tilde{s}, z)$. The degree of each $E_m(\tilde{s}, z)$ there is $\ell_e(m) = (2m + 1)n_1$ in $s$ (higher by $n_1$ than here) and degree $n_2 - m$ in $z$. It terminates with $E_{n_2}(\tilde{s}, z)$ of degree $2n_1(n_2 + 1)$. The final positivity test is applied again to a symmetric polynomial of degree $2n_1n_2$ that is obtained by dividing $E_{n_2}(\tilde{s}, z)$ by $E_0(\tilde{s}, 1)$. Thus, approximately, neglecting $O(n_{1,2}^5)$ counts, the test here and in [7] are of comparable cost of computation. However, a more detailed comparison reveals that the current test form wins by comparison with [7] in count of operations and simplicity of its setting.

It is also possible to similarly carry out a count of operation for the *scattering* tabular test in [13]. The matrices there do not possess any symmetry. Therefore convolutions have to be carried out fully. The count of operations for the method there can be shown to be approximately $\frac{4}{3}n_1^2n_2^4 + O(n_{1,2}^5)$, [8]. This indicates a 1.6 factor of advantage for very large $n$ (letting $n = n_1 = n_2$). Following an exact count of operations for the two tests shows (after setting into it again $n = n_1 = n_2$) that the cost ratio is higher than 2 for degree values of practical interest ($3 \leq n \leq 10$).

Clearly, the 2-D stability can be equally determined by carrying out the test for the 2-D polynomial with transposed coefficient matrix ($D \to D^t$). According to the above expression for the count of operations, it is preferable to apply the test to the 2-D polynomial with transposed coefficient matrix when $n_1 < n_2$.

### 5.3. *Numerical example*

For illustration, consider the polynomial used as an example in several papers [14], [16], [20, p. 129], [7]:

$$D(z_1, z_2) = [1 z_1^1 z_1^2] \begin{bmatrix} 0 & 0 & 0.2500 \\ 0 & 0.2500 & 0.5000 \\ 0.2500 & 0.5000 & 1.0000 \end{bmatrix} \begin{bmatrix} 1 \\ z_2^1 \\ z_2^2 \end{bmatrix}.$$

$D(z, 1) = D(1, z) = [0.2500\,0.7500\,1.7500]\mathbf{z}$ are easily determined to be stable. (The entries in parentheses reflect structural symmetry.)

$$M = \begin{bmatrix} 0 & 0 & 0.4375 \\ 0 & 0.4375 & 1.0625 \\ 0.4375 & 1.0625 & 2.1875 \\ 0.1875 & 0.4375 & 0.8750 \\ 0.0625 & 0.1250 & 0.2500 \end{bmatrix}$$

$$E_0 = \begin{bmatrix} 0.2500 & 0.1250 & 0.5000 \\ 0.8750 & 0.8750 & 1.2500 \\ 2.6250 & 2.1250 & (2.6250) \\ (1.2500) & (0.8750) & (0.8750) \\ (0.5000) & (0.1250) & (0.2500) \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0.2500 & 0.3750 \\ 0.8750 & 0.8750 \\ 1.7500 & (1.7500) \\ (0.8750) & (0.8750) \\ (0.3750) & (0.2500) \end{bmatrix}$$

Set m = 0: $g_1 = [0.0938, 0.5469, 2.1875, 4.5156, 6.7031, 5.1406, 2.6250, 0.7500, 0.1250]^t$ and $q_1 = [0.0938, 0.5469, 1.8594, 3.6094, 4.7969, (3.6094), (1.8594), (0.5469), (0.0938)]^t$. Obtain the right hand side numerator polynomial's matrix coefficient; in this case the matrix has one column [0.0547, 0.4336, 2.0840, 6.3281, 14.066, 22.301, 26.447, (22.301), (14.066), (6.3281), (2.0840), (0.4336), (0.0547)]$^t$. Obtain $q_0$ by summing the columns of $E_0$,

$$q_0 = [0.8750,\ 3.0000,\ 7.3750,\ (3.0000),\ (0.8750)]^t,$$

and deconvolve the numerator with $q_0$. The row size is reduced by $4(= 2n_1)$, and the result is

$$E_2 = \begin{bmatrix} 0.0625 \\ 0.2812 \\ 0.8906 \\ 1.5938 \\ 2.0781 \\ (1.5938) \\ (0.8906) \\ (0.2812) \\ (0.0625) \end{bmatrix}.$$

The degree $n_2 = 2$ of this example is too low to allow an $E_2$ with several columns and show that they all are obtained after deconvolution with $q_0$. A higher degree of $n_2$ is also needed to illustrate the division that stems from Lemma 3. It remains to examine whether $\epsilon_2(s) = \mathbf{s}^t E_2 \neq 0\ \forall s \in T$. This test may be carried out with the 1-D zero location test for real polynomials in [2]. Following the method there, the next stability table is formed.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.1250 | | 0.5625 | | 1.7813 | | 3.1875 | | 4.1562 | | (3.1875)... | |
| | −1.5625 | | −7.500 | | −20.250 | | −32.562 | | (−32.562) | | ... |
| | | 0.4375 | | 0.8125 | | 1.9375 | | 1.2500 | | (1.9375)... | |
| | | | 3.0357 | | 10.429 | | 21.179 | | (21.179) | | ... |
| | | | | 1.1279 | | 2.6176 | | 4.8544 | | (2.6176)... | |
| | | | | | −0.3478 | | −1.0684 | | (−1.0684) | | ... |
| | | | | | | 1.9755 | | 2.0760 | | (1.9755) | |
| | | | | | | | 0.3552 | | (0.3552) | | |
| | | | | | | | | 1.8750 | | | |

(Right halves of the symmetric rows are partly truncated.) According to the rules there, the information on zero location with respect to $T$ is extracted from the number of sign variations of the ordered sequence formed by the sum of the rows in this table:

$$Var\,\{15.468, -123.75, 7.6250, 69.286, 12.346, -2.8325, 6.0271, 0.7104, 1.8750\} = 4$$

4 sign variations means that the tested polynomial has 4 zeros in $V$. Being a symmetric polynomial of degree 8; it then also has 4 zeros in $U$ (the reciprocals of the zeros in $V$) and no zeros on $T$, i.e., $\epsilon_2(s) \neq 0$ for all $s \in T$. Therefore, the examined $D(z_1, z_2)$ is stable.

## 6. Concluding remarks

This paper has developed a method for testing the stability of 2-D discrete-time system polynomials. The test consists of the construction of a sequence of centrosymmetric polynomials or matrices (the *table*) and stability conditions posed on it. The stability conditions are posed on 1-D polynomials and require in the minimal form one stability test of degree $n_1$ (or $n_2$) and testing a symmetric polynomial of degree $2n_1n_2$ for no zeros (or positivity of the balanced polynomial) on $T$.

This test is not limited to real-valued $D$. It can also equally be used for 2-D polynomials with complex coefficients. The validity of this assertion becomes apparent from the following facts. The test was developed using a 1-D test for *complex* coefficient polynomials and using Lemma 1 which aslo holds for complex $D$. Then, the subsequent derivation used nowhere any assumption that limts it to a real $D$. In the complex case; the $^\sharp$ operation also includes conjugation. This symmetry and centrosymmetry stands for conjugate-symmetry and centro-Hermitian symmetry. For the common case of real $D$, the test is presented such that it encounters no complex number arithmetics.

The current test has many parallels with the immittance tabular test in [7], but there are also some important differences. The test here is based on the original complex 1-D stability test in [2], [3] whereas the test in [7]: is based on the modified 1-D test in [6]. Close examination of the two tests reveals that the current test requires a little less computation and poses the positivity conditions more directly than the other. Both tests may be regarded as the immittance counterpart of the scattering 2-D stability test of Hu and Jury [13]. The centrosymmetry of the matrices makes the immittance-type 2-D stability tests more efficient than jury's scattering-type 2-D test. The symmetry also makes the immittance tabular tests more stable numerically than their scattering counterpart. The latter property stems from the fact that because of the symmetry it suffices to carry out convolutions to only half of their full length and so reduce the accumulation of numerical error. More work on the 2-D stability testing problem that aims to reduce further cost of computation and increase numerical accuracy is currently in progress.

## References


[1] B. D. Anderson and E. I. Jury, Stability test for two-dimensional recursive filters, *IEEE Trans. Audio Elect.*, vol. AU-21, 366–372, 1973.

[2] Y. Bistritz, Zero location with respect to the unit circle of discrete-time linear system polynomials, *Proc. IEEE*, vol. 72, 1131–1142, Sept. 1984.

[3] Y. Bistritz, A circular stability test for general polynomials, *Systems Control Lett.*, vol. 7, 89–97, 1986.

[4] Y. Bistritz, Stability test for 2-D LSI systems via a unit-circle test for complex polynomials, in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 789–792, Seattle, WA, 1995.

[5] Y. Bistritz, Reflection on Schur-Cohn matrices and Jury-Marden tables and classification of related unit circle zero location criteria, *Circuits Systems Signal Process*, vol. 15, 111–136, 1996.

[6] Y. Bistritz, A modified unit-circle zero location test, *IEEE Trans. Circuits and Systems-I*, CAS-43, 472–475, 1996.

[7] Y. Bistritz, Stability testing of 2-D discrete linear system polynomials by a 2-D tabular form, *IEEE Trans. Circuits and Systems, Part I*, vol. 46, 666–676, June 1999.

[8] Y. Bistritz, On Jury's test for 2-D stability of discrete-time systems and its simplification by telepolation, *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, Switzerland, 2000, 567–570, to appear in ISCAS, 2000. (Journal paper version is also in preparation.)

[9] Y. Bistritz, H. Lev-Ari, and T. Kailath, Immittance type three-term Levinson and Schur recursions for quasi-Toeplitz complex Hermitian matrices, *SIAM J. Matrix Anal. Appl.*, vol. 12, 497–520, 1991.

[10] N. K. Bose, *Applied Multidimensional System Theory*, Van Nostrand Reinhold, Princeton, NJ, 1982.

[11] D. Dudgeon and R. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[12] D. M. Goodman, Some stability properties of two-dimensional shift-invariant digital filters, *IEEE Trans. Circuits and Systems*, vol. CAS-24, 201–208, 1977.

[13] X. Hu and E. I. Jury, On two-dimensional filter stability test, *IEEE Trans. Circuits and Systems*, vol. CAS-41, 457–462, 1994.

[14] T. S. Huang, Stability of two-dimensional recursive filters, *IEEE Trans. Audio Electro Acous-*

*tics*, vol. AU-20, 158–163, June 1972.

[15] E. I. Jury, Stability of multidimensional systems and related problems, in *Multidimensional Systems: Techniques and Applications*, S. G. Tzafestas, ed., Chap. 3, Marcel Dekker, New York, 1986 .

[16] E. I. Jury, Modified stability table for 2-D digital filter, *IEEE Trans. Circuits and Systems*, vol. CAS-35, 116–119, 1988.

[17] E. I. Jury, A note on the modified stability table for linear discrete time systems, *IEEE Trans. Circuits and Systems*, vol. CAS-38, 221–223, 1991.

[18] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[19] G. A. Maria and M. M. Fahmy, On the stability of two-dimensional digital filters, *IEEE Trans. Audio and Electroacoust.*, vol. AU-21, 470–472, 1973.

[20] B. T. O'Connor and T. S. Huang, Stability of general two-dimensional recursive digital filters, in *Two-dimensional Digital Signal Processing I : Linear Filters*, T. S. Huang, ed., Ch. 4, Springer-Verlag, Berlin, 1981.

[21] D. D. Siljak, Stability criteria for two-variable polynomials, *IEEE Trans. Circuits and Systems*, vol. CAS-22, 185–189, March 1975.

[22] M. G. Strintzis, Test of stability of multidimensional filters, *IEEE Trans. Circuits and Systems*, vol. CAS-24, 432–437, August 1977.