Stability Testing of Two-Dimensional Discrete Linear System Polynomials by a Two-Dimensional Tabular Form

Yuval Bistritz, *Senior Member, IEEE*

***Abstract*— A new test for determining whether a bivariate polynomial does not vanish in the closed exterior of the unit bicircle (is stable) is developed. A stable bivariate polynomial is the key for stability of two-dimensional (2-D) recursive linear discrete systems. The 2-D stability test stems from a modified stability test for one-dimensional (1-D) systems that has been developed by the author. It consists of a 2-D table, a sequence of centro-symmetric matrices, and a set of accompanying necessary and sufficient conditions for 2-D stability imposed on it. The 2-D table is constructed by a three-term recursion of these matrices or corresponding bivariate polynomials. The minimal set of necessary and sufficient conditions for stability consists of testing two univariate polynomial, one before and one after completing the table, for no zeros outside and no zeros on the unit circle, respectively. A larger set of useful conditions that are necessary for 2-D stability, and may indicate earlier instability, is also shown.**

***Index Terms*—Digital filters, discrete-time systems, linear systems, multidimensional systems, polynomials, stability.**

I. Introduction

An important issue in the design and analysis of two-dimensional (2-D) linear discrete systems is their stability. A 2-D system is considered stable (in the BIBO sense) if bounded input signals produce bounded output signals. It is the same definition that is used as well for the stability of one-dimensional (1-D) systems with the same purpose: to ensure a well-behaved system with predictable steady state. However, as with other issues in processing 2-D signals, testing stability of a 2-D system is more difficult because the simplicity of the mathematics used for 1-D systems is absent for higher dimensional systems. For 1-D polynomials the fundamental theorem of algebra states that any polynomial of degree n can be factored as a product of n polynomials of degree one. This theorem does not hold for multivariate polynomials. Consequently, polynomials obtained by the Z transform of a 2-D difference equation cannot be factored in terms of lower degree polynomials. In 1-D systems, stability testing amounts to examining the location of the poles of the transfer function with respect to the unit circle. Stability may be determined either by numerical calculation of these poles or by using one of several available algebraic 1-D stability tests that determine stability in a finite number of operations without determination of the numerical values of the poles. For 2-D systems, the former choice is not available because the poles are, in general, not a countable set of points, but surfaces in a four-dimensional (4-D) space that are hard to localize and are not confined to any closed subset of the space. In spite of these difficulties, and some further reservations that will be mentioned later, it emerges that the main problem in stability determination of a 2-D discrete system may be stated in a manner that looks like an anticipated generalization of the 1-D stability problem.

Problem Statement: Given a 2-D (bivariate) polynomial determine whether it does not vanish in the closed exterior of the unit bicircle, viz.

$$D(z_1, z_2) = \sum_{i=0}^{n_1} \sum_{k=0}^{n_2} d_{i,k} z_1^i z_2^k \neq 0, \quad \forall (z_1, z_2) \in \overline{V} \times \overline{V}. \tag{1}$$

$T = \{z\colon |z| = 1\}, U = \{z\colon |z| < 1\}, V = \{z\colon |z| > 1\}$ are used to denote the unit circle, its interior, and its exterior, respectively, and the bar denotes closure $\overline{V} = V \cup T$.

The paper proposes a new test for solving the stated problem. A 2-D polynomial $D(z_1, z_2)$ that satisfies (1) will be called stable. Similarly, a 1-D (univariate) polynomial is called stable if

$$D(z) = \sum_{k=0}^{n} d_k z^k \neq 0, \qquad \forall z \in \overline{V}. \tag{2}$$

The latter similarity between the conditions for 2-D and 1-D stable polynomials is useful for the extension of 1-D stability tests to the 2-D case, but it hides the fact that 2-D stable polynomials relate to stability of 2-D systems in a more complicated manner than in the 1-D case. We note also that other notations for a discrete stable 2-D polynomial are also used in the literature and require (simple) conversion to the form (1). We shall dwell briefly on these subject in Section I-B. A more comprehensive coverage on this stability problem and its multidimensional systems background is available in [1]–[5].

The proposed method to solve the problem is algebraic, namely, unlike numerical or graphical stability tests, it aims at providing a definite answer, in a finite number of arithmetic operation, to whether (1) holds. The test consists of a sequence of centro-symmetric matrices, referred as a 2-D table, that is constructed by a three-term recursion of 2-D polynomials and

Manuscript received October 10, 1995; revised November 12, 1998. This paper was recommended by Associate Editor A. Kummert.

The author is with the Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel.

Publisher Item Identifier S 1057-7122(99)04744-3.

of a few accompanying conditions on 1-D polynomials, that may be examined by unit circle zero location tests. The new method is based on the method in [6] for determining the location of zeros of a 1-D polynomial with respect to the unit circle, which is one of the so called immittance counterparts to the class of Marden–Jury tables of the Schur–Cohn algorithms for stability testing [7]. Other immittance algorithms offer alternative solutions to additional classical (scattering) signal processing algorithms related to the Schur–Cohn algorithm [8].

The first tabular stability test for 2-D stability was proposed by Maria and Fahmy [9]. It was based on an early form of the Marden–Jury 1-D stability table. Anderson and Jury proposed to solve the problem by a polynomial Schur–Cohn matrix [10]. Subsequently, Siljak showed that for testing 2-D stability via positive definiteness of the Schur–Cohn polynomial matrix over the unit circle, it suffices to determine its definiteness at a single point and positivity over T of its determinant polynomial (saving similar examination of lower principle minor polynomials) [11]. Jury designed a modified tabular 1-D stability test that produces explicitly the principal minors of the Schur–Cohn matrix [12], [13] and, as such, it is capable of combining the manageability of a tabular test with the simplification in computation introduced by Siljak. More recently, Hu and Jury [14] improved this test by removing from its implementation redundant factors.

The paper develops the new test in two stages in an order that follows the evolution of Marden–Jury type 2-D stability tables in the literature. In the first stage, a recursion for the construction of a sequence of matrices called the F table is presented and a certain set of conditions that are necessary and sufficient for stability are obtained for it. This set of stability conditions requires the examination of several 1-D polynomials, one for each matrix in the sequence, and determining whether they do not vanish (are positive) on T. (This point of development approximately parallels the Maria and Fahmy test.) Afterwards it is shown that it suffices to examine only the last of these 1-D polynomials (comparable to the single positivity test simplification in the scattering approach [11]–[13]). In the second stage, it is first shown that the number of rows in the matrices of the F table is higher than necessary and a reduced size table form called the E table is obtained. Next, the stability conditions for the F table are shown to hold also for the E table. The decrease in the number of rows in the E table reduces significantly the cost of computation of the final form of the table and its single positivity test. The transition from the F table to the reduced E table parallels the improvement that Hu and Jury contributed in [14] to previous 2-D tests based on the 1-D stability test of Jury. The contribution in this paper differs in scope in that it goes all the way from the 1-D stability test in [6] to the immittance counterpart of the result in [14]. It also differs from previous approaches in that, currently, the single positivity stability condition is proved directly from intrinsic properties of the underlying recursion. Each stage in the current development compares well in cost of computation with respective scattering counterpart 2-D stability tests, because the immittance approach exhibits certain structural symmetries that may be used to compute less entries in the 2-D table.

The derivation of the new 2-D stability test is shown in an instructive manner without using appendices for details and proofs. Preceding the E table with the F table is useful, not only to follow the evolution of corresponding scattering 2-D tests, but also to organize the derivation in a tractable order. This mode of presentation achieves clarity without lengthening the paper, because proofs for theorems for the E table follow from proofs of corresponding F table theorem after a brief explanation of the needed adjustments.

A. Notation

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

$$\begin{aligned}P(s,z)&=\sum_{k=0}^{n_2} p_k(s)z^k\\&=[p_0(s),p_1(s),\cdots,p_{n_2}(s)]\boldsymbol{z}=\boldsymbol{s}^tP\boldsymbol{z}.\end{aligned}$$

Here, p_k is the $k+1$th column of P and $p_k(s)=\boldsymbol{s}^tp_k$ 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 $p_k=[p_{0,k},p_{1,k},\cdots p_{n_1,k}]^t$, but mainly we shall manipulate vectors as a whole and act on columns of matrices. Superscript $\rightleftharpoons$ will denote (conjugate) reversion, defined for a matrix and a vector by

$$P^{\rightleftharpoons}=JP^*J \quad p^{\rightleftharpoons}=Jp^*$$

respectively, where J denotes the reversion matrix with ones on the main antidiagonal and zeros elsewhere and * denotes complex conjugate.

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

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

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

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

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

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

and it will represent extraction of a factor $f(s)$ common to all the polynomials $g_k(s)$.

Notation such as $[0,G]$ or $[G,0,0]$ will denote pre- or post-padding of the columns of the matrix G by a shown number of columns of zeros of the same length.

During the process of developing the new method it will be useful to think of the coefficient matrix P of $P(s,z)$ as associated with the next function

$$P(\tilde{s},z)=s^{-n_1/2}P(s,z)=\sum_{k=0}^{n_2}p_k(\tilde{s})z^k=\tilde{\boldsymbol{s}}^tP\boldsymbol{z} \quad (3)$$

where $\tilde{\boldsymbol{s}}:=[s^{-m/2},s^{-(m/2-1)},\cdots,s^{(m/2-1)},s^{m/2}]^t$ (of a length determined by the context, e.g., $m=n_1$ here) and $\tilde{s}$ as a function argument denotes power series to equal extent in each of the two variables (s^{-1},s) or $(s^{-1/2},s^{1/2})$ (for m even or odd, respectively). A function such as $p_k(\tilde{s})$ is called a balanced polynomial.

We shall construct for the polynomial $D(z_1,z_2)$ a sequence of matrices $\{E_m, m=-1,0,\cdots,n(=n_2)\}$ which are centro-symmetric $E_m^{\rightleftharpoons}=E_m$. These matrices may be linked to either $E_m(s,z)$ or $E_m(\tilde{s},z)$ by the above convention. The 2-D polynomial $E_m(s,z)$ will be of degree n_2-m in z and of a certain degree $\ell_e(m)$ in s which increases with m in a manner that will be discussed later. When reference is made to columns or entries of a matrix or vector that is member in a sequence, the sequential index m will be set in brackets and precede other indexes. For example $E_m(\tilde{s},z)=[e_{[m]0}(\tilde{s}),e_{[m]1}(\tilde{s}),\cdots,e_{[m]n-m}(\tilde{s})]\boldsymbol{z}=\tilde{\boldsymbol{s}}^t[e_{[m]0}, e_{[m]1},\cdots,e_{[m]n-m}]\boldsymbol{z}$ where $e_{[m]k}=[e_{[m]0,k},e_{[m]1,k},\cdots,e_{[m]\ell_e(m),k}]^t$ is the $k+1$th column of E_m.

B. On 2-D Stability and Stable 2-D Polynomials

As was mentioned already, the relation of 2-D stable polynomials to stable 2-D discrete systems is more complicated than in the 1-D case, see [1]–[5] for detailed coverage on the subject. We give here a brief account on these differences and on the conversion between 2-D stable polynomials, as defined in (1), and alternative conventions that are also in use in the literature.

Assume a system transfer function of a recursive discrete 2-D system or filter with first quadrant support consisting of the ratio of two finite degree 2-D polynomials (a system with different wedge support can be transformed to the first quadrant by a simple linear mapping without affecting its stability): say $N(z_1,z_2)$ and $D(z_1,z_2)$. Assume that these two polynomials are coprime, namely, they have no common factors (except a constant or a linear phase term). For a 1-D system with a transfer function, say $N(z)/D(z)$, if the numerator and denominator polynomials are coprime, they may not have common zeros and thus stability of the system is determined solely by stability of $D(z)$. In the 2-D case, because 2-D polynomials are not factorable in general, the coprime polynomials $N(z_1,z_2)$ and $D(z_1,z_2)$ may still have zero surfaces that intersect on T^2 at values that are then called nonessential singularities of the second kind (NSSK) of $N(z_1,z_2)/D(z_1,z_2)$. It was shown in [15] that NSSK may stabilize a system with an unstable $D(z_1,z_2)$. Thus, in a strict mathematical sense a stable $D(z_1,z_2)$ is a sufficient, but not necessary, condition for 2-D stability. It becomes a necessary condition for stability if the system is assumed to have no NSSK. However, as was said in [1], it appears that in practice, recursive filter design algorithms will virtually never produce NSSK on T^2 and, at the same time, satisfying $D(z_1,z_2)\neq 0$ on $\overline{V}^2-T^2$. Also, a system with $D(z_1,z_2)$ that vanishes on T^2, but is stabilized by the numerator vanishing there as well, represents a situation of zero stability margin that is not acceptable as stable in practice. So, in a practical sense, a stable $D(z_1,z_2)$ is also a necessary condition for a system's stability.

There is no uniquely agreed convention to define the stable 2-D polynomial in the literature. For 1-D systems, the Z transform is defined mainly in negative powers of z. So a 1-D polynomial $A(z)=[1,z^{-1},\cdots,z^{-n}]a$ is stable if $A(z)\neq 0$ $\forall z\in\overline{V}$. Using a similarly negative convention for the Z transform in two variables, a 2-D polynomial $A(z_1,z_2)$ should be defined as stable if it has the form and satisfies the relation

$$\begin{aligned}A(z_1,z_2)&=[1,z_1^{-1},\cdots,z_1^{-n_1}]A[1,z_2^{-1},\cdots,z_2^{-n_2}]^t\\ &\neq 0 \quad \forall(z_1,z_2)\in\overline{V}\times\overline{V}.\end{aligned} \quad (4)$$

This convention is used, for example, in [3] and [5]. Clearly, $A(z_1,z_2)$ is stable if and only if $D(z_1,z_2)$ with $D=JAJ$ is stable, and $A(z)$ is stable if and only if $D(z)$ with coefficient vector $d=Ja$ is stable. Conversion from $A(z_1,z_2)$ to $D(z_1,z_2)$ happens when both polynomials of a rational transfer function obtained by the Z transform, defined with negative powers, are multiplied by a common zero-phase factor to get a denominator polynomial of positive powers. Several texts on stability of multidimensional systems reach the convenience of dealing with polynomials in positive powers simply by using the Z transform defined in positive powers of its variables, cf. [1], [2], [4]. In this case, a polynomial $B(z_1,z_2)$ is created by associating the above matrix A with positive powers of z_1 and z_2. Thus, $B(z_1,z_2)$ is stable if it has the form and satisfies the relation

$$\begin{aligned}B(z_1,z_2)&=[1,z_1,\cdots,z_1^{n_1}]A[1,z_2,\cdots,z_2^{n_2}]^t\\ &\neq 0\forall(z_1,z_2)\in\overline{U}\times\overline{U}.\end{aligned} \quad (5)$$

Evidently, $B(z_1,z_2)$ is stable if and only if $A(z_1,z_2):=B(z_1^{-1},z_2^{-1})$ is stable. In summary, to test the condition (4) or (5), the test in its current form has to be applied to $D=JAJ$. Alternatively, the test (shown in Section I-C) can be adjusted to incorporate A by carrying into it the substitution $D\rightarrow A^{\rightleftharpoons}$, which involves only a simple change in the initiation of the table construction.

C. Preview of the 2-D Stability Test

Using the notation in Section I-A, we summarize here the 2-D stability testing procedure that is developed in this paper.

In the following, steps in brackets are optional steps, i.e., they are suggested but may be skipped. Exit marks a point at which the algorithm may be interrupted with a $D(z_1,z_2)$ is not stable conclusion. An illustration by a numerical example will be provided in Section V after completing the development of the test.

The Proposed 2-D Stability Test: To test whether $D(z_1, z_2)$ is stable, i.e., whether (1) holds, proceed as follows.

Step 1: Pre-Examinations.
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, such as $D(z, z)$ or $D(s_o, z)$ and $D(z, s_o)$ at $s_o \in \overline{V}$, e.g., $s_o = \infty, -1$, and exit if any of them is not stable.]

Step 2: 2-D Table Construction.
Obtain the sequence of centro-symmetric matrices $E_m, m = -1, 0, 1, \cdots, n_2$ ($E_m = E_m^{\sharp}$ is of size $(n_1(2m+1)) \times (n_2 - m)$ for $m \geq 0$)

$$\begin{aligned} E_{-1} &= [0, D - D^{\sharp}] - [D - D^{\sharp}, 0] \\ E_0 &= D + D^{\sharp} \\ \epsilon_0 &= \sum_{k=0}^{n} e_{[0]k}, \quad q_{-1} := 1 \end{aligned} \tag{25$'$}$$

For $m = 0, \cdots, n-1$ $(n := n_2)$ do:
[Optionally (for $m \geq 1$) compute $\epsilon_m = \Sigma_{k=0}^{n-m} e_{[m]k}, \hat{\epsilon}_m = \epsilon_m / \epsilon_0$.
Test whether $\boldsymbol{s}^t \hat{\epsilon}_m \neq 0 \ \forall s \in T$
False—exit, True—continue.]

$$\begin{aligned} q_m &= e_{[m]0} * e_{[m]0}^{\sharp}; \quad g_m = e_{[m-1]0} * e_{[m]0}^{\sharp} \\ \tilde{E}_m &= g_m * [E_m, 0] \\ [0, E_{m+1}, 0] &= (\tilde{E}_m + \tilde{E}_m^{\sharp} - q_m * E_{m-1}) / q_{m-1}. \end{aligned} \tag{26$'$}$$

Step 3: Post-Examination.

$$\hat{\epsilon}_n = E_n / \epsilon_0.$$

Test whether $\hat{\epsilon}_n(s) = \boldsymbol{s}^t \hat{\epsilon}_n \neq 0 \ \forall s \in T$
False—exit, True—$D(z_1, z_2)$ is stable.
The condition in Step 3 as well as similar optional conditions $\hat{\epsilon}_m(s) = \boldsymbol{s}^t \hat{\epsilon}_m \neq 0 \ \forall s \in T$ for $m < n$ in Step 2 may be replaced by the condition $\hat{\epsilon}_m(\tilde{s}) = \tilde{\boldsymbol{s}}^t \hat{\epsilon}_m > 0$ $\forall s \in T$ and are therefore also referred to as positivity tests. All the mentioned 1-D polynomial tests may be carried out algebraically by unit circle zero location tests, including [6] that underlies the current 2-D test.

The paper is organized as follows. The Section II brings two auxiliary results: a simplification to condition (1) and a modification for the stability test in [6]. Section III derives a preliminary form for the test, the F table, and its stability conditions. Section IV derives the final form of the proposed 2-D test, the E table, and its stability conditions. Section V makes some comments on implementation of the new 2-D test and brings a numerical illustration. The paper ends with some concluding remarks.

II. Auxiliary Results

This section cites first a simplification of the condition (1) that is the starting point of most algebraic methods for testing it. Then it modifies the 1-D stability test in [6] to a form that is more suitable for extension to 2-D stability testing.

A. Huang–Strintzis Stability Conditions

Lemma 1: (Huang–Strintzis.) $D(z_1, z_2)$ is stable if and only if

a)
$$D(z, a) \neq 0, \qquad \forall z \in \overline{V} \text{ and some } a \in \overline{V} \tag{6}$$
b)
$$D(s, z) \neq 0, \qquad \forall (s, z) \in T \times \overline{V}. \tag{7}$$

This Lemma was introduced to the field by Huang for $a = \infty$ [16], [1] and in its above form by Strintzis [17]. It states that the search over $\overline{V} \times \overline{V}$ (a 4-D subspace of the bivariate complex plane) in (1) may be replaced by a search of (z_1, z_2) over just $T \times \overline{V}$ (a 3-D subspace). Other simplifying forms of stability conditions of this kind are also known [4], [18], but they do not seem to offer true extra computational merit. It is possible, for example, to relate our derivation to the Decarlo–Strintzis simplification which confines the search to $D(s_1, s_2) \neq 0 \ \forall (s_1, s_2) \in T \times T$ cf. [3] or [5]. This condition represents a search of just a 2-D subspace, but its examination requires the same effort as condition (7). It is desirable to choose a real a because then no complex arithmetic is introduced for a real D (the common case). We fix our choice to $a = 1$ that integrates nicely with the special role that $z = 1$ plays in our immittance stability conditions, and we shall not state further the existence of alternatives.

B. Modified 1-D Stability Test

Our starting point attempts to combine Lemma 1 with the 1-D stability test in [6]. The algorithm in [6] uses a three-term recursion with a multiplier (denoted there by δ_m) that is obtained by division of two numbers. It is desirable to obtain a version of the test that is free of this division in order to circumvent dealing with rational functions of s in its intended application. Such a division-free version of [6] is derived below. An apparent advantage of the following modification is that when used in conjunction with Lemma 1, it admits manipulation of only polynomials. Some additional advantages of this modification will be noted after its presentation.

Consider a 1-D polynomial

$$p(z) = \sum_{k=0}^{n} p_k z^k, \quad \mathrm{Re}\{p(1)\} \neq 0 \tag{8}$$

where p_k are complex scalars and $\mathrm{Re}\{\cdot\}$ denotes real part of.

Algorithm 1: Division-Free 1-D Table. Obtain for the polynomial (8) the following sequence of polynomials $\{f_m(z) = \Sigma_{k=0}^{n-m} f_{[m]k} z^k, \ m = -1, 0, 1, \cdots, n\}$ and scalars $\{\phi_m, m = 0, 1, \cdots, n\}$

i) *Initiation:*

$$\begin{aligned} f_{-1}(z) &= (z-1)(p(z) - p^{\sharp}(z)) \\ f_0(z) &= p(z) + p^{\sharp}(z) \end{aligned} \tag{9}$$

and $\phi_0 := f_0(1)$ $(= 2\mathrm{Re}\{p(1)\} \neq 0$ by assumption)

ii) *Recursion:* For $m = 0, \cdots, n-1$:

$$\begin{aligned} zf_{m+1}(z) = &(f_{[m-1]0}f^*_{[m]0} + f^*_{[m-1]0}f_{[m]0}z)f_m(z) \\ &- f_{[m]0}f^*_{[m]0}f_{m-1}(z) \end{aligned} \tag{10}$$

and $\phi_{m+1} := f_{m+1}(1)$

Theorem 1: 1-D Stability Conditions. Assume Algorithm 1 is applied to $p(z)$ (8). $p(z)$ is stable if and only if

$$\frac{\phi_m}{\phi_0} > 0, \qquad m = 1, \cdots, n. \tag{11}$$

Proof: Denote the sequence for $p(z)$ in [6] by $\{t_m(z)\}$ (rather than $\{F_m(z)\}$ used for it in there). Compare the recursions here with the recursion in [6] to realize that the relation between the sequences $\{f_m(z)\}$ here and $\{t_m(z)\}$ there is $f_m(z) = \psi_m t_m(z)$ with $\psi_m = |f_{[m-1]0}|^2\psi_{m-2} = |t_{[m-1]0}|^2\psi^2_{m-1}\psi_{m-2}$, $m \geq 1, \psi_{-1} = 1, \psi_0 = 1$. The current necessary and sufficient conditions follow from corresponding stability conditions in [6] via the fact that all the ψ_m are real and positive. ■

Remark 1: The polynomials $\{f_m(z), m = -1, 0, 1, \cdots, n\}$ produced by Algorithm 1 are conjugate symmetric. $f_m(z)$ is of degree $n-m$ and $f_{[m]n-m-i} = f^*_{[m]i}, i = 0, 1, \cdots, n-m$. The normal conditions in [6] transform here to the condition that all $f_{[m]0} \neq 0$. Normal conditions remain necessary conditions for stability and the condition $f_{[m]0} = 0$ implies and is detected by a subsequent violation of (11). However, differing from [6], the situation $f_{[m]0} = 0$ does not affect the recursion because division by $f_{[m]0}$ has been eliminated.

Remark 2: In the forthcoming 2-D stability testing task, $p(z)$ and hence all $f_m(z)$ will assume coefficients dependent on $s \in T$. The fact that the recursion moves intactly through $f_{[m]0} = 0$ situations is valuable for this application because it will save tests as to whether $f_{[m]0}$ as a polynomials in s vanishes on T. Another useful property of the test in this context (shared also with the original form) is that no requirement on $p_n \neq 0$ is posed. A polynomial of degree n with $p_n = 0$ has (at least one) zero at infinity which implies it is not stable. Were this test used directly for testing 1-D stability, observation of $p_n = 0$ is sufficient to determine the polynomial as not stable (or the lower degree polynomial with not vanishing leading coefficient may be taken, if [6] is used to determine the distribution with respect to the unit circle of the remaining zeros). For the current use, the fact that the test is not obstructed by vanishing leading coefficients is again valuable. The implied instability shows as a violation of the stated stability condition. Indeed, the following relation is easily derived $\phi_1 = 2(|p_n|^2 - |p_0|^2)\phi_0$ from which it is seen that if $p_n = 0$ then $\phi_1/\phi_0 \leq 0$, i.e., (11) is not satisfied.

III. Preliminary 2-D Stability Test Form

Our intention is to apply the division-free 1-D stability test of the previous section to test (7). It is noticed that (7) holds if and only if

$$D(\tilde{s}, z) \neq 0 \quad \forall (s, z) \in T \times \overline{V}. \tag{12}$$

As a consequence, it is possible to test (7) by applying the division-free 1-D stability test to $D(\tilde{s}, z)$ rather than to $D(s, z)$. The result of application of Algorithm 1 to $D(\tilde{s}, z)$, regarding it as a 1-D polynomial in z with balanced polynomial coefficients dependent on $s \in T$, is described below in Algorithm 2. This algorithm associates D with a sequence of matrices $\{F_m, m = -1, 0, \cdots, n = n_2\}$ that we call the F table. The algorithm below uses polynomial notation in which the matrices appear as coefficients of the sequence of polynomials. Polynomial interpretation is needed for derivation of the method. It is possible afterwards to convert such an algorithm into operation on vectors for more obvious programming in a matrix environment, as demonstrated for the final form of the 2-D table.

The advantage of using $D(\tilde{s}, z)$ rather than $D(s, z)$ follows from the fact that the complex conjugate of balanced polynomials for values $s \in T$ retains the length of their coefficient vectors. Using the 1-D stability with $D(s, z)$ instead, would have doubled the row sizes of F_{-1} and F_0 and, as a consequence, all the matrices in the sequence associated with D.

A. Construction of the F Table

Algorithm 2: The F Table (Preliminary Table Form). Construct for $D(z_1, z_2)$ a sequence of polynomials $\{F_m(\tilde{s}, z) = \sigma_{k=0}^{n-m} f_{[m]k}(\tilde{s})z^k, m = -1, 0, 1, \cdots, n(= n_2)\}$ by the following recursion.

i) *Initiation:*

$$\begin{aligned} F_{-1}(\tilde{s}, z) &= (z-1)(D(\tilde{s}, z) - D^{\tilde{*}}(\tilde{s}, z)) \\ F_0(\tilde{s}, z) &= D(\tilde{s}, z) + D^{\tilde{*}}(\tilde{s}, z). \end{aligned} \tag{13}$$

ii) *Recursion:* For $m = 0, 1, \cdots, n-1$ obtain $F_{m+1}(\tilde{s}, z)$ by

$$\begin{aligned} h_m(\tilde{s}) &= f_{[m-1]0}(\tilde{s})f^{\tilde{*}}_{[m]0}(\tilde{s}) \\ r_m(\tilde{s}) &= f_{[m]0}(\tilde{s})f^{\tilde{*}}_{[m]0}(\tilde{s}) \\ zF_{m+1}(\tilde{s}, z) &= h_m(\tilde{s})F_m(\tilde{s}, z) + h^{\tilde{*}}_m(\tilde{s})zF_m(\tilde{s}, z) \\ &\quad - r_m(\tilde{s})F_{m-1}(\tilde{s}, z). \end{aligned} \tag{14}$$

B. Stability Conditions for the F Table

In order to supplement Algorithm 2 with stability conditions we associate it with two auxiliary sequences of (conjugate) symmetric (balanced) polynomials

$$\begin{aligned} \varphi_m(\tilde{s}) &:= F_m(\tilde{s}, 1) = \sum_{k=0}^{n-m} f_{[m]k}(\tilde{s}), \\ \hat{\varphi}_m(\tilde{s}) &= \frac{\varphi_m(\tilde{s})}{\varphi_0(\tilde{s})} \quad m = -1, 0, \cdots, n. \end{aligned} \tag{15}$$

A recursion that the sequence $\{\varphi_m(\tilde{s})\}$ obeys is obtained by setting $z = 1$ in (14)

$$\varphi_{m+1}(\tilde{s}) = h^r_m(\tilde{s})\varphi_m(\tilde{s}) - r_m(\tilde{s})\varphi_{m-1}(\tilde{s}) \tag{16}$$

for $m = 0, \cdots, n-1$ where

$$\begin{aligned} h^r_m(\tilde{s}) &:= h_m(\tilde{s}) + h^{\tilde{*}}_m(\tilde{s}), \\ \varphi_0(\tilde{s}) &= F(\tilde{s}, 1),\ \varphi_{-1}(\tilde{s}) = 0. \end{aligned}$$

The second sequence $\{\hat{\varphi}_m(\tilde{s})\}$ obeys a similar recursion for $m = 0, \cdots, n-1$

$$\hat{\varphi}_{m+1}(\tilde{s}) = h_m^r(\tilde{s})\hat{\varphi}_m(\tilde{s}) - r_m(\tilde{s})\hat{\varphi}_{m-1}(\tilde{s}) \tag{17}$$

with different initiation $\hat{\varphi}_0(\tilde{s}) = 1, \hat{\varphi}_{-1} = 0$. It becomes apparent that $\varphi_0(\tilde{s})$ is a factor of all $\varphi_m(\tilde{s})$ and that $\hat{\varphi}_m(s)$ are polynomials. Furthermore, all $\varphi_m(\tilde{s})$ and all $\hat{\varphi}_m(\tilde{s})$ are (conjugate) symmetric balanced polynomials $\hat{\varphi}_m^{\sim} = \hat{\varphi}_m$. A polynomial $\varphi_m(s) = \boldsymbol{s}^t\varphi_m$, with such symmetry, is characterized by unit circle or reciprocal pairs of zero (i.e., if $\varphi_m(s_o) = 0$ then also $\varphi_m(1/s_o^*) = 0$). Another consequence of this symmetry is that all $\varphi_m(\tilde{s})$ and $\hat{\varphi}_m(\tilde{s})$ are real $\forall s \in T$.

The polynomials $\hat{\varphi}_m(\tilde{s})$ celebrate in the following stability theorems. The recursions they obey are brought because they will be used in the proofs of these theorems. Note that $h_m^r(\tilde{s})$ and $r_m(\tilde{s})$ in these recursions require the polynomials $\{F_m(\tilde{s}, z)\}$ and once the latter are available it is simpler to obtain $\{\varphi_m(\tilde{s})\}$ from their definition (15).

Theorem 2: F Table's Stability Conditions. $D(z_1, z_2)$ is stable if and only if the following conditions (a) and (b) or (b′) hold.

a) $D(z, 1) \neq 0 \ \forall z \in \overline{V}$

b) $\hat{\varphi}_m(\tilde{s}) > 0 \ \forall s \in T$ for $m = 1, \cdots, n$

b′)

i) $D(1, z) \neq 0 \ \forall z \in \overline{V}$

ii) $\hat{\varphi}_m(s) \neq 0 \ \forall s \in T$ for $m = 1, \cdots, n$

where $\hat{\varphi}_m(\tilde{s})$ are obtained from the F table of $D(z_1, z_2)$.

Proof: It is necessary to show that condition b) here and in the Lemma 1 are equivalent. Assume $D(s, z) \neq 0 \ \forall (s, z) \in T \times \overline{V}$ holds, then condition (12) holds. Assume first values of $s \in T$ such that $\varphi_0(\tilde{s}) = 2\text{Re}\{D(\tilde{s}, 1)\} \neq 0$. For each such $s \in T$ Algorithm 2 is an implementation of Algorithm 1 for $P_s(z) = D(\tilde{s}, z) = \Sigma \ d_m(\tilde{s})z^m$. Therefore, by Theorem 1, $\hat{\varphi}_m(\tilde{s}) > 0$ for all $m > 0$ if and only if $D(\tilde{s}, z)$ is 1-D stable as a polynomial in z, i.e., $D(\tilde{s}, z) \neq 0 \ \forall z \in \overline{V}$.

Next, consider values $s_o \in T$ for which $\varphi_0(\tilde{s}_o) = 0$. They need special attention because they represent values for which the requirement in (8) does not hold. Note that they may occur whether or not $D(z_1, z_2)$ is stable. (In fact for odd n $\varphi_0(\tilde{s})$ must vanish at $s = -1$.) Assume that $\varphi_0(\tilde{s}_o) = 0$ for a certain $s_o \in T$. We have to show that the stated conditions are necessary and sufficient for stability of $P_{s_o}(z) = D(\tilde{s}_0, z)$ as well. Let us focus on a vicinity $\mathcal{V}_{s_o} \subset T$ such that s_o is in its interior and $\varphi_0(s) \neq 0$ for $s_o \neq s \in \mathcal{V}_{s_o}$. Consider the part of the root location of $P_s(z) = D(\tilde{s}, z) = 0$ that corresponds to $s \in \mathcal{V}_{s_o}$. As has been shown already, $\hat{\varphi}_m(\tilde{s}) > 0 \ \forall m > 0$ for all $s_o \neq s \in \mathcal{V}_{s_o}$ if and only if the roots of $P_s(z)$ lie in U. Therefore no roots may be in V also for $P_{s_o}(z)$. It remains to negate the possibility that a branch of this mapping may touch T, i.e., that $P_{s_o}(z)$ has zeros on T. In the context of the underlying 1-D stability test, zeros on T were discussed under the category of structural or type I singularity [6] and they were shown to imply, and be implied by, a later $f_{m_o}(z) \equiv 0$ for some $m_o > 0$. Since all $\hat{\varphi}_m(\tilde{s})$ are continuous for all $s \in \mathcal{V}_{s_o}$, this latter situation is possible if and only if $\hat{\varphi}_{m_o}(\tilde{s}_o) = 0$ for that m_o which is inconsistent with condition b).

Finally, we show that conditions b) and b′) are equivalent. Clearly, the condition $\hat{\varphi}_m(\tilde{s}) > 0$ on T is equivalent to $\hat{\varphi}_m(\tilde{s}) \neq 0$ on T plus positivity at one point on T, say $\hat{\varphi}_m(1) > 0$. Therefore the conditions in b) are replaceable by the next pair of conditions: i′) $\hat{\varphi}_m(1) > 0, m = 1, \cdots, n$ and ii′) $\hat{\varphi}_m(\tilde{s}) \neq 0, m = 1, \cdots, n$. i′) holds if and only if $D(1, z)$ is stable by Theorem 1, and ii′) and ii) are obviously equivalent. ■

Remark 3: Condition b′) is more practical than b). The 1-D stability testing of $D(1, z)$ is relatively simple and may precede the construction of the table. If it is found not to be stable then $D(z_1, z_2)$ is not stable and the construction of the 2-D table is not needed. Testing by algebraic means the condition $\hat{\varphi}_m(\tilde{s}) > 0$ or the condition $\hat{\varphi}_m(s) \neq 0$ on T (the latter is not real valued on T) is of equal complexity. Therefore, in the following theorems we leave stability of $D(1, z)$ as part of the requirement, even if it appears to be adding an extra condition to the number of acompanying conditions that we strive to reduce. We shall refer to both forms b) and ii) as positivity conditions.

Here is a further characterization for the recursions (16) and (17) that we shall need at a later stage.

Corollary from Theorem 2: If $D(z_1, z_2)$ is stable then

i) $h_m^r(\tilde{s}) > 0 \ \forall s \in T$ for $m = 0, \cdots, n-1$.

ii) $r_m(\tilde{s}) > 0 \ \forall s \in T$ for $m = 0, \cdots, n-1$.

Proof: If condition b) of Theorem 2 holds then condition i) is implied via (17) because, by definition, $r_m(\tilde{s}) \geq 0$ on T. In turn, i) implies ii). Indeed, a negation of ii) means that $r_m(\tilde{s}_o) = 0$ for some $s_o \in T$. This implies $f_{[m]0}(\tilde{s}_o) = 0$, which in turn implies $h_m^r(s_o) = 0$: a contradiction to i). ■

C. Reduced Stability Conditions

Let $\ell_f(m)$ be the degree of all $f_{[m]k}(s)$ and of $\varphi_m(s) = F_m(s, 1)$ (i.e., the row size of F_m is $\ell_f(m)+1$). The three-term recursion (14) induces the relation

$$\ell_f(m+2) - 2\ell_f(m+1) - \ell_f(m) = 0. \tag{18}$$

A closed form expression for $\ell_f(m)$ can be easily obtained by solving this difference equation for the initial conditions $\ell_f(-1) = \ell_f(0) = n_1$. It suffices to realize that the solution is a linear combination of the two modes $\lambda_{1,2} = 1 \pm \sqrt{2}$ that increases exponentially with m due to the λ^m term with $\lambda = 1 + \sqrt{2}(\approx 2.414)$. As a consequence, the table size increases exponentially with $n = n_2$ and positivity tests of polynomials of rapidly increasing degrees are required. The next theorem offers some simplification. Accordingly, while all n_2 positivity conditions are necessary for stability, only the last positivity test must be examined.

Theorem 3: Refined F Table's Stability Conditions. $D(z_1, z_2)$ is stable if and only if conditions i), ii), and iii) or iii′) hold.

i) $D(z, 1) \neq 0 \ \forall z \in \overline{V}$

ii) $D(1, z) \neq 0 \ \forall z \in \overline{V}$

iii) $\hat{\varphi}_n(s) \neq 0 \ \forall s \in T$

iii′) $\hat{\varphi}_n(\tilde{s}) > 0 \ \forall s \in T$

where $\hat{\varphi}_n$ is obtained from the F table of $D(z_1, z_2)$.

Proof: These conditions are necessary because they form a subset of the conditions of Theorem 2.

To prove sufficiency, we proceed to show that the current conditions i)–iii) imply the larger set of sufficiency conditions in Theorem 2 $\varphi_m(\tilde{s}) \neq 0$ for $m < n$ as well. In the following argument we shall refer to the closeness of a point $s_o \in T$ to $s = 1$ through the distance of its real part $s_o^R := \mathrm{Re}\{s_o\}$ to $s = 1$.

Assume that conditions i)–iii) hold but that, nevertheless, there exists one (or several) $\hat{\varphi}_m(\tilde{s}), m < n$ that vanish at one (or several) values of $s \in T$. By Theorem 1, condition ii) implies that $\hat{\varphi}_m(\tilde{s}) > 0$ at $s = 1$ for all $m \geq 0$. Let $\hat{\varphi}_k(\tilde{s})$ be the earliest $\hat{\varphi}_m(\tilde{s})$ (i.e., k is the least m) such that $\hat{\varphi}_k(\tilde{s}) = 0$ for some $s \in T$. And let s_1 denote the zero on T of $\hat{\varphi}_k(\tilde{s})$ closest to $s = 1$. $\hat{\varphi}_k(\tilde{s}_1) = 0$ implies, via (17),

$$\hat{\varphi}_{k+1}(\tilde{s}_1) = -r_k(\tilde{s}_1)\hat{\varphi}_{k-1}(\tilde{s}_1) \leq 0$$

because $r_k(\tilde{s}) \geq 0 \ \forall s \in T$ (by its definition) and $\hat{\varphi}_{k-1}(\tilde{s}_1) > 0$ (by the assumed choice of k). Therefore, $\hat{\varphi}_{k+1}(\tilde{s})$ must vanish for some $s \in T$ whose real part is in the interval in $[s_1^R, 1)$. Let s_2 be the root of $\hat{\varphi}_{k+1}(\tilde{s}) = 0$ on T closest to $s = 1$, i.e., with maximal s_2^R, then $s_1^R \leq s_2^R < 1$. Repeating this reasoning $n_o = n - k + 1$ times implies that $\hat{\varphi}_n(\tilde{s})$ must vanish for some $s \in T$ with the real part in a subinterval $[s_{n_o}^R, 1)$ where $s_{n_o}^R < 1$. This conclusion is in contradiction to assumption iii). Therefore, conditions i)–iii) imply $\hat{\varphi}_m(\tilde{s}) \neq 0 \ \forall s \in T$ for all $m < n$ as well. The sufficiency of i)–iii) for stability of $D(z_1, z_2)$ follows now from the sufficiency of the conditions in Theorem 2. ■

IV. Final Form of the 2-D Stability Test

It turns out that the polynomials $F_m(s, z)$ produced by Algorithm 2 are separable into two polynomials

$$F_m(s, z) = \alpha_m(s) E_m(s, z) \tag{19}$$

where each $\alpha_m(s)$ is a polynomial in s only, which is therefore a common factor for all the polynomial coefficients $f_{[m]k}(s)$ in the presentation of $F_m(s, z)$ as $F_m(s, z) = \Sigma_{k=0}^{n-m} f_{[m]k}(s) z^k$.

This section proves this property and characterizes it. Afterwards, an algorithm to obtain the sequence of E_m matrices, referred to as the E table, is presented. The row size of each E_m is lower than the corresponding size of F_m by an amount equal to the degree of $\alpha_m(s)$. The row sizes of the new sequence of matrices will then be shown to increase only linearly with m. Finally, stability conditions for this E table are established. It will be shown that the reduction in size of the table does not complicate the simple form of the stability conditions found so far for the F table. Therefore, the eliminated $\alpha_m(s)$ are and will be called redundant factors. The simplified E table and its stability conditions will constitute the new 2-D stability test proposed in this paper.

A. Redundant Factors

The next Lemma exposes the above mentioned common $\tilde{s}$ factors and features their rapid accumulation.

Lemma 2: Consider a sequence $\{F_m(\tilde{s}, z)\}$ produced by the recursion (14).

1) 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 the recursion (14) $g_{[1]0}(\tilde{s}) g^{\bar{\prime}}_{[1]0}(\tilde{s})$ is a factor of $G_3(\tilde{s}, z) = \sum g_{[3]i}(\tilde{s}) z^i$. Namely, $g_{[1]0}(\tilde{s}) g^{\bar{\prime}}_{[1]0}(\tilde{s})$ divides, with no remainder, each $g_{[3]i}(\tilde{s})$.
2) If $f(\tilde{s})$ is a factor of $F_m(\tilde{s}, z), m \geq 0$, then it is a factor of all subsequent $F_{m+i}(\tilde{s}, z) \ i \geq 1$.

Proof: To prove property a), write two legitimate consecutive matrices of the F table as

$$G_0 = [g_{[0]0}, g_{[0]1}, \cdots, g^{\bar{\prime}}_{[0]1}, g^{\bar{\prime}}_{[0]0}] \tag{20}$$

$$G_1 = [g_{[1]0}, g_{[1]1}, \cdots, g^{\bar{\prime}}_{[1]1}, g^{\bar{\prime}}_{[1]0}] \tag{21}$$

where G_1 has one less column than G_0. The next two matrices that recursion (14) will then generate are

$$\begin{aligned}[0, G_2, 0] &= g_{[0]0} * g^{\bar{\prime}}_{[1]0}[G_1, 0] + g^{\bar{\prime}}_{[0]0} * g_{[1]0}[0, G_1] \\ &\quad - g_{[1]0} * g^{\bar{\prime}}_{[1]0} G_0 \\ &= [0, g_{[2]0}, g_{[2]1}, \cdots, g^{\bar{\prime}}_{[2]1}, g^{\bar{\prime}}_{[2]0}, 0]\end{aligned} \tag{22}$$

$$\begin{aligned}&[0, 0, G_3, 0, 0] \\ &\quad = g_{[1]0} * g^{\bar{\prime}}_{[2]0}[0, G_2, 0, 0] \\ &\qquad + g^{\bar{\prime}}_{[1]0} * g_{[2]0}[0, 0, G_2, 0] \\ &\qquad + g_{[2]0} * g^{\bar{\prime}}_{[2]0} * [0, G_1, 0]\end{aligned} \tag{23}$$

where columns of zeros are padded to bring all matrices to the column size of G_0. For the proof, G_3 has to be expressed in terms of columns of G_0 and G_1. For simplicity, we may drop from the resulting sum of terms, terms that are already seen to contain the factor $g_{[1]0} * g^{\bar{\prime}}_{[1]0}$. The justification follows from fact that the three-term recursion has the property that any term that contains a certain factor, contributes to subsequent polynomials terms that also contain that factor.

Thus, we replace (22) with

$$[0, G_2, 0] \mapsto g_{[0]0} * g^{\bar{\prime}}_{[1]0}[G_1, 0] + g^{\bar{\prime}}_{[0]0} * g_{[1]0}[0, G_1] \tag{24}$$

where we use the symbol $\mapsto$ to mean that the right-hand side (r.h.s) is what remains after evaluating the left hand side and dropping terms seen to contain the factor $g_{[1]0} * g^{\bar{\prime}}_{[1]0}$. By comparing the second column in the two sides of the above expression, obtain

$$g_{[2]0} \mapsto g_{[0]0} * g^{\bar{\prime}}_{[1]0} * g_{[1]1} + g^{\bar{\prime}}_{[0]0} * g_{[1]0} * g_{[1]0}.$$

Use this expression for $g_{[2]0}$ (and its conjugate reversion) to prepare the next two auxiliary results

$$g_{[1]0} * g^{\bar{\prime}}_{[2]0} \mapsto g^{\bar{\prime}}_{[0]0} * g_{[1]0} * g_{[1]0} * g^{\bar{\prime}}_{[1]1}$$

$$\begin{aligned}g_{[2]0} * g^{\bar{\prime}}_{[2]0} \mapsto\ & g_{[0]0} * g_{[0]0} * g^{\bar{\prime}}_{[1]0} * g^{\bar{\prime}}_{[1]0} * g^{\bar{\prime}}_{[1]0} * g_{[1]1} \\ &+ g^{\bar{\prime}}_{[0]0} * g^{\bar{\prime}}_{[0]0} * g_{[1]0} * g_{[1]0} * g_{[1]0} * g^{\bar{\prime}}_{[1]1}.\end{aligned}$$

Substitute the first of them and (24) into the first term in the r.h.s of (23) to obtain for it alone

$$\begin{aligned}g_{[1]0} * g^{\bar{\prime}}_{[2]0}[0, G_2, 0, 0] \mapsto\ & g^{\bar{\prime}}_{[0]0} * g^{\bar{\prime}}_{[0]0} * g_{[1]0} * g_{[1]0} * g_{[1]0} \\ &* g^{\bar{\prime}}_{[1]1}[0, G_1, 0].\end{aligned}$$

Add to the second term in the r.h.s of (23) the conjugate reversion of this expression. Then, substitute the auxiliary result for $g_{[2]0} * g^{\tilde{}}_{[2]0}$ into the third term in the right-hand side of (23). Adding up these three terms, the third exactly wipes out the first two. In conclusion

$$[0, 0, G_3, 0, 0] \mapsto 0.$$

It follows from here, together with the convention assigned to $\mapsto$, that G_3 is composed of a sum of terms, each of which contains the factor $g_{[1]0} * g^{\tilde{}}_{[1]0}$ plus terms that sum up to zero. This concludes the proof of a).

Property b) follows at once from the observation that in the three-term recursion (14) each new $F_{m+i}(\tilde{s}, z)$ is a combination of terms that either contain $F_{m+i-1}(\tilde{s}, z)$ as a whole or contain the term $f_{[m+i-1]0}(\tilde{s})$. ■

It follows from Lemma 2 that each $F_{k+2}(\tilde{s}, z)$, $k = 0, 1, \cdots, n-2$ is divisible by each of the factors $f_{[i]0}(\tilde{s}) f^{\tilde{}}_{[i]0}(\tilde{s}), i = 0, \cdots, k$ and that these factors accumulate and increase their multiplicity as the recursion goes on.

B. Construction of the E Table

One obvious way to eliminate the common factors spotted by Lemma 2 is to divide them out after the F table has been completed. This approach has some merit, as it can be shown to reduce the degree of the polynomials whose positivity is to be examined, but it does not elevate the main computational burden presented by the construction of the F table. In terms of both computation and numerical stability, it is more desirable to produce directly the reduced-size E table. A closer inspection on Lemma 2 reveals that it has not just exposed the existence of common factors, but it also indicates the mechanism that may be used to remove them as soon as they are created. The next algorithm implements this insight and derives directly the sequence $\{E_m(\tilde{s}, z), m = -1, 0, \cdots, n\}$. Namely, each $E_m(\tilde{s}, z)$ produced by Algorithm 3 corresponds to $F_m(\tilde{s}, z)$ stripped from all common s factors revealed by Lemma 2.

Algorithm 3: The E Table (Final Table Form). Construct for $D(z_1, z_2)$ a sequence of polynomials $\{E_m(\tilde{s}, z) = \Sigma_{k=0}^{n-m} e_{[m]k}(\tilde{s}) z^k \; m = -1, 0, \cdots, n (= n_2)\}$ as follows:

i) *Initiation:*

$$\begin{aligned} E_{-1}(\tilde{s}, z) &= (z-1)(D(\tilde{s}, z) - D^{\tilde{}}(\tilde{s}, z)) \\ E_0(\tilde{s}, z) &= D(\tilde{s}, z) + D^{\tilde{}}(\tilde{s}, z). \end{aligned} \tag{25}$$

ii) *Recursion*: For $m = 0, 1, \cdots, n-1$ obtain $E_{m+1}(\tilde{s}, z)$ by:

$$\begin{aligned} g_m(\tilde{s}) &= e_{[m-1]0}(\tilde{s}) e^{\tilde{}}_{[m]0}(\tilde{s}) \\ q_m(\tilde{s}) &= e_{[m]0}(\tilde{s}) e^{\tilde{}}_{[m]0}(\tilde{s}) \end{aligned}$$

$$zE_{m+1}(\tilde{s}, z) = \frac{g_m(\tilde{s}) E_m(\tilde{s}, z) + g^{\tilde{}}_m(\tilde{s}) z E_m(\tilde{s}, z) - q_m(\tilde{s}) E_{m-1}(\tilde{s}, z)}{q_{m-1}(\tilde{s})} \tag{26}$$

where $q_{-1}(\tilde{s}) = 1$.

C. Stability Conditions

We want to obtain stability conditions for the E table from corresponding conditions for the F table. For this, we first explore the exact relation between the sequences $\{F_m(\tilde{s}, z)\}$ and $\{E_m(\tilde{s}, z)\}$. It is clear that they are related by a relation of the form (19) and that $\alpha_m(s)$ are (conjugate) symmetric $\alpha^{\tilde{}}_m(s) = \alpha_m(s)$. So the balanced polynomials $\alpha_m(\tilde{s})$ are real $\forall s \in T$ and the degree of $\alpha_m(s)$ represents the amount of row reduction achieved by moving from F_m to E_m. Substitution of the relation (19) into one table recursion, and then comparing it to the other table recursion, reveals the next recursive rules for the $\alpha_m(s)$'s. The initiation is $\alpha_0(\tilde{s}) = \alpha_1(\tilde{s}) = 1$ and then they may be determined, given the sequence $\{E_m(s, z)\}$, by

$$\alpha_{m+1}(\tilde{s}) = \alpha_m^2(\tilde{s}) \alpha_{m-1}(\tilde{s}) q_{m-1}(\tilde{s}), \qquad m \geq 1 \tag{27}$$

or, given the sequence $\{F_m(s, z)\}$, by

$$\alpha_{m+1}(\tilde{s}) = \frac{\alpha_m^2(\tilde{s})}{\alpha_{m-1}(\tilde{s})} r_{m-1}(\tilde{s}), \qquad m \geq 1. \tag{28}$$

The next relation that may also be obtained

$$\alpha_{m+1}(\tilde{s}) = \prod_{i=0}^{m-1} r_i^{m-i}(\tilde{s}), \qquad m \geq 1 \tag{29}$$

demonstrates the rapid accumulation of common factors. We also define for the E table the (conjugate) symmetric (balanced) polynomial sequences $\{\epsilon_m(\tilde{s})\}$ and $\{\hat{\epsilon}_m(\tilde{s})\}, m = -1, 0, \cdots, n$

$$\epsilon_m(\tilde{s}) = E_m(\tilde{s}, 1), \quad \hat{\epsilon}_m(\tilde{s}) = \frac{\epsilon_m(\tilde{s})}{\epsilon_0(\tilde{s})}. \tag{30}$$

They correspond to the F table polynomials $\varphi_m(\tilde{s})$ and $\hat{\varphi}_m(\tilde{s})$. The sequence $\{\hat{\epsilon}_m(\tilde{s})\}$ obeys the recursion

$$\hat{\epsilon}_{m+1}(\tilde{s}) = \frac{g_m^r(\tilde{s}) \hat{\epsilon}_m(\tilde{s}) - q_m(\tilde{s}) \hat{\epsilon}_{m-1}(\tilde{s})}{q_{m-1}(\tilde{s})} \tag{31}$$

$m = 0, \cdots, n-1$ where

$$g_m^r(\tilde{s}) := g_m(\tilde{s}) + g^{\tilde{}}_m(\tilde{s}), \quad \hat{\epsilon}_0(\tilde{s}) = 1, \quad \hat{\epsilon}_{-1} = 0$$

as may be verified by setting $z = 1$ in (26). [The sequence $\{\epsilon_m(\tilde{s})\}$ obeys a similar recursion with the initiation $\epsilon_{-1} = 0$, $\epsilon_0(\tilde{s}) = E(\tilde{s}, 1)$.] Once again, the recursion requires the construction of the E table and is brought to serve the proofs of the following stability conditions.

The next theorem is the E table counterpart of Theorem 2.

Theorem 4: E Table's Stability Condition. $D(z_1, z_2)$ is stable if and only if conditions a) and b) or b′) hold.

a) $D(z, 1) \neq 0 \; \forall z \in \overline{V}G$.

b) $\hat{\epsilon}_m(\tilde{s}) > 0 \; \forall s \in T$ for $m = 1, \cdots, n$.

b′)

i) $D(1, z) \neq 0 \; \forall z \in \overline{V}$.

ii) $\hat{\epsilon}_m(s) \neq 0 \; \forall s \in T$ for $m = 1, \cdots, n$.

where $\{\hat{\epsilon}_m\}$ are obtained from the E table of $D(z_1, z_2)$.

Proof: If $D(z_1, z_2)$ is stable then by Theorem 2 and its corollary all $\hat{\varphi}_m(\tilde{s})$ and $r_m(\tilde{s})$ are positive on T. Therefore, all $\alpha_m(\tilde{s}) > 0$ on T by (28) or (29). Conditions b) and b′) follow now from their Theorem 2 counterparts via (19).

Conversely, assume the conditions a) and b) or b′) hold. If so, we show that $q_m(\tilde{s}) > 0$ $m = 0, \cdots, n-1$ on T is implied where, by definition, all $q_m(\tilde{s}) \geq 0$ on T. Assume the contrary. Namely, $q_m(\tilde{s_o}) = 0$ for some m and $s_o \in T$. This implies $e_{[m]0}(s_o) = 0$ which then implies $g_m(s_o) = 0$. Together they imply, via (31), that $\hat{\epsilon}_{m+1}(\tilde{s_o}) = 0$ in contradiction to b) and b′). Therefore, $q_m(\tilde{s}) > 0$ for $m = 0, \cdots, n-1$ on T. Then, by (27) all $\alpha_m(\tilde{s}) > 0$ on T and b) implies that all $\hat{\varphi}_m(\tilde{s}) > 0$ on T. Consequently, the assumed conditions imply that $D(z_1, z_2)$ is stable by the sufficiency part of Theorem 2. ∎

The last and main theorem states that a more concise set of stability conditions, similar to Theorem 3, holds also for the reduced size E table.

Theorem 5: Main Theorem. $D(z_1, z_2)$ is stable if and only if conditions i), ii), and iii) or iii′) hold.

i) $D(z, 1) \neq 0 \ \forall z \in \overline{V}$
ii) $D(1, z) \neq 0 \ \forall z \in \overline{V}$
iii) $\hat{\epsilon}_n(s) \neq 0 \ \forall s \in T$
iii′) $\hat{\epsilon}_n(\tilde{s}) > 0 \ \forall s \in T$

where $\hat{\epsilon}_n$ is obtained from the E table of $D(z_1, z_2)$.

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

A proof that i), ii), iii), or iii′) imply the larger set of conditions a), b), or b′) in Theorem 4 can be carried out by extending the proof of sufficiency for Theorem 3 from the F table to the E table. To this end it suffices to show that the difference between the F table and the E table recursions does not affect the argument on the migration of hypothetical zeros of $\hat{\varphi}_m(\tilde{s})$ on T downward in the recursion, used to prove Theorem 3. A repetition of the argument there with $\hat{\epsilon}_m(\tilde{s})$ replacing $\hat{\varphi}_m(\tilde{s})$ is possible after the following observation. A zero on T of $\hat{\epsilon}_k(\tilde{s})$ affects immediately the next $\hat{\epsilon}_{k+1}(\tilde{s})$. In contrast, a factor $q_m(\tilde{s})$ that is formed at the recursion step $m = k$ divides the right-hand side of (26) only at step $m = k+2$. Consequently, a (hypothetical) zero on T of $\hat{\epsilon}_k(\tilde{s})$ is passed to subsequent $\hat{\epsilon}_m(\tilde{s})$'s before the division by $q_m(\tilde{s})$, added in the E table, has a chance to cancel it out. Thus, the line of the proof for Theorem 3 can be repeated to also prove the sufficiency part of the current theorem by showing that any $\hat{\epsilon}_m(\tilde{s}_o) = 0$ for an $s_o \in T$ and an $m < n$ contradicts condition iii). ∎

This theorem proves at last that the common factors that were eliminated in the process of replacing the F table by the E table are indeed redundant. Their removal reduces significantly the computation cost of the table's construction, without complicating the associated stability conditions. As a matter of fact, the single positivity condition of the last theorem benefits in itself from the general reduction in size, because the polynomial to be tested has a lower degree. Let $\ell_f(m) + 1$ and $\ell_e(m) + 1$ be the row sizes for F_m and E_m, respectively. It was shown in (18) that $\ell_f(m)$ increases exponentially with m. For comparison, the recursion (26) implies for $\ell_e(m)$ the equation

$$\ell_e(m+2) - 2\ell_e(m+1) + \ell_e(m) = 0 \qquad (32)$$

whose solution for the initial values $\ell_e(0) = n_1$ and $\ell_e(1) = 3n_1$ is $\ell_e(m) = (2m+1)n_1$. So, $\ell_e(m)$ increases only linearly with m. The amount of saving in computation for the E table compared to the F table increases rapidly with n_2, cf (29). All $\hat{\epsilon}_m(s)$ are also of lower degrees than corresponding $\hat{\varphi}_m(s)$. In particular, $\hat{\epsilon}_n(s)$, the only polynomial obtained from the table that has to be examined, is a symmetric polynomial of degree $2n_1n_2$. The stepping down from exponential to linear orders is remarkable already at low values of n_1 and n_2. As an illustration, for $n_1 = n_2 = 4$ the F table row polynomials have degrees $\{\ell_f(m)\}_0^4 = \{4, 12, 28, 68, 164\}$ and $\hat{\varphi}_n(s)$ is of degree 160, while the E table has the degrees $\{\ell_e(m)\}_0^4 = \{4, 12, 20, 28, 36\}$ and the degree of $\hat{\epsilon}_n(s)$ is 32.

V. Implementation Issues and a Numerical Illustration

A possible procedure for performing the proposed 2-D stability test has been summarized in Section I-C. The proposed procedure there may now be recognized as consisting of a translation of Algorithm 3 into a more matricial form (using the conversion and notation in Section I-A) plus the stability conditions of Theorem 5. It also incorporates, as optional steps, 1-D stability conditions and positivity conditions that, according to Theorem 4, are necessary for 2-D stability. Matrix presentation makes the programming of the test more transparent in matrix-oriented programming languages. For a full matrix presentation, the convolution or deconvolution may be written as multiplication of one vector by a lower triangular Toeplitz matrix defined by the other vector or by its simple-to-calculate inverse matrix. (Matlab has built-in fast routines for convolution and deconvolution.) The optional 1-D polynomial tests may prove useful for determining stability constraints on literal parameters and/or may save computation in repetitive application by detecting earlier that the 2-D polynomial is not stable.

The examination of a 1-D polynomial of degree n for stability or for no zeros on T can be carried out in order-n^2 operations using either classical Marden–Jury and Schur–Cohn classes of tests [7], including [12] and [13], or the immittance tests [19], [20], including the test on which the current 2-D stability test was based [6]. The use of the Schur–Cohn and Marden–Jury (SCMJ) stability tests requires adaptation to the singular situation caused by the symmetry of the tested polynomial and extension to the zero location, with respect to the unit circle not widely available for all versions [7]. The mentioned immittance test references handle the singular cases and the count of zeros inside, on, and outside the unit circle, as currently required. They also exceed the efficiency of all possible alternative versions in the SCMJ class of methods by factors of two to four. Specifically, the methods in [19] and [6] require $0.25n^2 + O(n)$ multiplications for testing a 1-D real polynomial of degree n for no zeros in $\overline{V}$ or on T. Corresponding versions for complex 1-D polynomials are available in [20] and [6].

TABLE I
ZERO LOCATION TABLE FOR $\hat{\epsilon}_2(s)$

0.5000		2.2500		7.1250		12.750		16.625		(12.750)	...
	−6.2500		−30.000		−81.000		−130.25		(−130.25)	...	
		1.7500		3.2500		7.7500		5.0000		(7.7500)	...
			12.143		41.714		84.714		(84.714)	...	
				4.5118		10.471		19.418		(10.471)	...
					−1.3911		−4.2783		(−4.2783)	...	
						7.9021		8.3041		(7.9021)	
							1.4207		(1.4207)		
								7.5000			

The test has been described such that a table of $n = n_2$ matrices is built for $D(z_1, z_2)$. It is always possible to let $n = n_1$ by preceding the test with the replacement $D \rightarrow D^t$. To reduce computation, n should be chosen as the lower of n_1 and n_2. A computationally efficient implementation of the proposed 2-D stability test should also exploit the symmetries in the arrays to compute and handle only half of their entries. In the following numerical illustration, entries that become available by structural symmetries are put in parentheses.

A. Numerical Example

For illustration, consider the polynomial $D(z_1, z_2)$ used as an example in several papers [1, p. 129], [16], [12]

$$D = \begin{bmatrix} 0.0000 & 0.0000 & 0.2500 \\ 0.0000 & 0.2500 & 0.5000 \\ 0.2500 & 0.5000 & 1.0000 \end{bmatrix}.$$

$D(z, 1) = D(1, z) = [0.2500,\ 0.7500,\ 1.7500]\boldsymbol{z}$ are easily determined to be stable

$$E_{-1} = \begin{bmatrix} 1.0000 & -0.5000 & (-0.5000) & (0.0000) \\ 0.5000 & -0.5000 & (-0.5000) & (0.5000) \\ 0.0000 & -0.5000 & (-0.5000) & (1.0000) \end{bmatrix}$$

$$E_0 = \begin{bmatrix} 1.0000 & 0.5000 & (0.5000) \\ 0.5000 & 0.5000 & (0.5000) \\ 0.5000 & (0.5000) & (1.0000) \end{bmatrix}$$

$$\epsilon_0(s) = E_0(s, 1) = [2.000, 1.500, (2.000)]\boldsymbol{s}$$

$$E_1 = \begin{bmatrix} 0.5000 & (0.5000) \\ 1.7500 & (1.5000) \\ 4.1250 & (3.7500) \\ 4.3750 & (4.3750) \\ 3.7500 & (4.1250) \\ 1.5000 & (1.7500) \\ 0.5000 & (0.5000) \end{bmatrix}.$$

The elimination of redundant factors appears first in the next recursion step. For this low $n_2 = 2$ case, the next E_2 is already the last matrix in the 2-D table and has only one column. Therefore, the low degree of the example inhibits a proper illustration of the fact that the eliminated factor is common to all columns of matrices from E_2 and on.

To obtain q_0 take the first column of E_0, $e_{[0]0} = [1.000, 0.500, 0.500]^t$ and form $q_0 = e_{[0]0} * \tilde{e}_{[0]0} = [0.5000, 0.7500, 1.5000, (0.7500), (0.5000)]\mathbf{s}$. Obtain the right-hand side of (26) $[0.2500,\ 1.6875, 7.3750, 21.46, 48.102, 82.781, 115.02,\ 127.27,\ (115.02),\ (82.781),\ (48.102),\ (21.469), (7.3750), (1.6875),\ (0.2500)]^t$. Then divide it by (deconvolve it with) q_o to obtain

$$E_2 = \begin{bmatrix} 0.5000 \\ 2.6250 \\ 9.3125 \\ 20.344 \\ 33.312 \\ 37.969 \\ (33.312) \\ (20.344) \\ (9.3125) \\ (2.6250) \\ (0.5000) \end{bmatrix}.$$

For the positivity test obtain $\hat{\epsilon}_2(s) = \boldsymbol{s}^t E_2 / \epsilon_0(s) = [0.2500,\ 1.1250,\ 3.5625, 6.3750,\ 8.3125,\ (6.375),\ (3.5625), (1.1250),\ (0.2500)]\boldsymbol{s}$. It remains to examine the condition $\hat{\epsilon}_2(s) \neq 0\ \forall s \in T$. We do this by the 1-D zero location test for real polynomials in [19]. Following the procedure there, obtain for $\hat{\epsilon}_2(s)$ the stability table in Table I. (Uncompleted rows contain entries that form mirror reflection of their left half side.) According to the rules there, the information on zero location with respect to T is contained in the number of sign variations of the ordered sequence formed by the sum of the rows in this table

$$\begin{aligned} \mathrm{Var}\{&61.875, -495.00, 30.500, 277.14, 49.382, -11.330, \\ &24.108, 2.8415, 7.5000\} = 4. \end{aligned}$$

Four sign variations mean that the tested polynomial has four zeros in V. Being a symmetric polynomial of degree eight, it has then also four zeros in U (the reciprocals of the zeros in V) and therefore no zeros on T, i.e., $\hat{\epsilon}_2(s) \neq 0\ \forall s \in T$. It is concluded that the examined $D(z_1, z_2)$ is stable.

VI. CONCLUDING REMARKS

The paper has developed a new method for testing stability of 2-D discrete system polynomials. It constructs for a 2-D system polynomial a 2-D table, a sequence of matrices (or 2-D polynomials), in a manner similar to the way that a 1-D tabular stability test associates the 1-D system polynomial with a sequence of row vectors (or 1-D polynomials). The algorithm for the construction of the proposed 2-D table has a simple recursive form that is readily implemented in a matrix-oriented environment, and the stability conditions for a 2-D polynomial of degree (n_1, n_2) require in their minimal form one 1-D stability test of degree n_1 or n_2 before the table is started and, after its completion, a test of one symmetric polynomial of degree $2n_1n_2$ for having no zeros on the unit circle. In contrast to the previous 2-D tabular tests that are based on the SCJM 1-D tests, and obey two-term recursions, the current test is based on an immittance type 1-D stability test and involves a three-term recursion of centro-symmetric matrices. This symmetry allows the actual computation of only half of the entries of the arrays. Unlike the common approach in 2-D tabular stability tests to seek relations between 1-D stability tests and the Schur-Cohn matrix minors in order to obtain extensions to 2-D stability tests with a single positivity test condition [12]–[14], [21], the single positivity test arises currently from intrinsic properties of the three-term 2-D polynomials recursion with no reference to extraneous relationships. Connections between the underlying 1-D test and the Schur-Cohn test are tractable [8], [6], but they provide a more complex, if not impenetrable, route to discover the simplicity of the current 2-D test. The current approach also enabled us to obtain a single positivity stability condition for other immittance 2-D stability tests whose underlying 1-D stability test is related in an even more complicated manner to the Schur-Cohn minors [22], [23]. The relative advantage of different 2-D stability tests and the possibility of lowering further their computational cost are subjects for further study.

It is worth noting that the current method for testing the condition (1) may be used also for the complex valued coefficient 2-D polynomial. The validity of the test for a complex D follows from the facts that the method was developed from a 1-D test for complex coefficient polynomials and that Lemma 1 is not restricted to real D. For complex D, the E_m's are complex and exhibit symmetry with respect to reversion plus complex conjugate. In the more ordinary application, when D is real, the test involves only real arithmetic and arrays.

REFERENCES

[1] 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. Berlin: Springer-Verlag, 1981.

[2] N. K. Bose, *Applied Multidimensional System Theory*. New York: Van Nostrand Reinhold, 1982.

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

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

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

[6] Y. Bistritz, "A modified unit circle zero location test," *IEEE Trans. Circuits Syst. I*, vol. 43, pp. 472–475, June 1996.

[7] 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, pp. 111–136, 1996.

[8] 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, pp. 497–520, 1991.

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

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

[11] D. D. Siljak, "Stability criteria for two-variable polynomials," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 185–189, Mar. 1975.

[12] E. I. Jury, "Modified stability table for 2-D digital filter," *IEEE Trans. Circuits Syst.*, vol. CAS-35, pp. 116–119, Jan. 1988.

[13] ——, "A note on the modified stability table for linear discrete time system," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 221–223, 1991.

[14] X. Hu and E. I. Jury, "On two-dimensional filter stability test," *IEEE Trans. Circuits Syst.*, vol. 41, pp. 457–462, July 1994.

[15] D. M. Goodman, "Some stability properties of two-dimensional shift-invariant digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 201–208, Apr. 1977.

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

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

[18] P. Delsarte, Y. Genin, and Y. Camp, "A simple proof of Rudin's multivariable stability theorem," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 701–704, Dec. 1980.

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

[20] Y. Bistritz, "A circular stability test for general polynomials," *Syst. Control Lett.*, vol. 7, pp. 89–97, 1986.

[21] K. Premaratne and E. I. Jury, "On the Bistritz Tabular form and its relationship with the Schur-Cohn minors and inner determinants," *J. Franklin Inst.*, vol. 330, pp. 165–182, 1993.

[22] Y. Bistritz, "An immittance-type stability test for two-dimensional digital filters," in *Proc. 28th Asilomar Conf. Signals, Systems, Computers*, 1994, pp. 918–922.

[23] ——, "Stability test for 2-D LSI systems via a unit circle test for complex polynomials," *in Proc. 1995 Int. Symp. Circuits Systems*, Seattle, WA, pp. 789–792, 1995.

Yuval Bistritz (S'79–M'83–SM'87) received the B.Sc. degree in physics and the M.Sc. and Ph.D. degrees in electrical engineering, from Tel Aviv University, Tel Aviv, Israel, in 1973, 1978, and 1983, respectively.

He served in the Israeli Defense Force from 1972 to 1975 and held a research engineering industrial position from 1975 to 1979. From 1979 to 1984 he held various assistant and teaching positions in the department of Electrical Engineering, Tel Aviv University. From 1984 to 1986 he was a Research Scholar in the Information System Laboratory, Stanford University, Stanford, CA, developing fast signal processing algorithms. From 1986 to 1987 he was with AT&T Bell Laboratories, Murray Hill, NJ, working on speech processing algorithms. Since 1987 he has been a Faculty Member in the Department of Electrical Engineering Systems, Tel Aviv University. His research interests are in the area of system and signal processing theory.