Stability Testing of 2-D Discrete Linear Systems by Telepolation of an Immittance-Type Tabular Test

Yuval Bistritz, *Senior Member, IEEE*

***Abstract*—A new procedure for deciding whether a bivariate (two-dimensional, 2-D) polynomial with real or complex coefficients does not vanish in the closed exterior of the unit bi-circle (is "2-D stable") is presented. It simplifies a recent immittance-type tabular stability test for 2-D discrete-time systems that creates for a polynomial of degree (n_1, n_2) a sequence of n_2 (or n_1) centro-symmetric 2-D polynomials (the "2-D table") and requires the testing of only one last one dimensional (1-D) symmetric polynomial of degree $2n_1n_2$ for no zeros on the unit circle. It is shown that it is possible to bring forth (to "*tele*scope") the last polynomials by inter*polation* without the construction of the 2-D table. The new 2-D stability test requires an apparently unprecedentedly low count of arithmetic operations. It also shows that stability of a 2-D polynomial of degree (n_1, n_2) is completely determined by $n_1n_2 + 1$ stability tests (of specific form) of 1-D polynomials of degrees n_1 or n_2 for the real case (or $2n_1n_2 + 1$ polynomials in the complex cases).**

***Index Terms*—Discrete-time systems, immittance algorithms, multidimensional digital filters, multidimensional systems, stability, stability criteria.**

I. Introduction

STABILITY of two-dimensional (2-D) linear discrete-time (shift-invariant) systems arises in many applications. It is required for the design of digital filters and processing of image, seismic, radar and other types of data, in multimedia, geography, communication, medicine and more fields. The key for testing stability of 2-D discrete systems, and the subject of this paper, is to determine whether a 2-D (bivariate) polynomial has no zeros in the closed exterior of the unit bi-circle. Background on stability of multidimensional systems and related issues is available in several texts, including [1]–[4]. These texts also contain comprehensive lists of references to earlier solutions for this problem. More recent contributions to this problem include [5]–[12], that will referenced again later in this work, and references there in.

This paper will present a new algebraic procedure that solves this problem in a very low count of arithmetic operations. The new procedure profits on the advantages of a recent immittance-type tabular 2-D stability test proposed for this problem in [13], [14] and simplifies it into an even more efficient 2-D stability test. The above tabular test builds for a polynomial of degree (n_1, n_2) a sequence of n_2 (or n_1) 2-D polynomials or matrices (the 2-D table) with certain symmetry. It then requires to check whether the table's last polynomial, a symmetric one-dimensional (1-D) polynomial of degree $2n_1n_2$, has no zeros on the unit-circle. The complexity of this tabular test is $O(n^6)$ (assuming $n = n_1 = n_2$ for convenience). It turns out that this complexity is dominated by the effort required for the construction of the 2-D table. The complexity of testing the last 1-D polynomial is only $O(n^4)$. The new stability test avoids the construction of the table and shows that it is possible instead to *tele*scopes (bring forth) the last 1-D polynomial of the 2-D table by inter*polation.* This approach, hence called *telepolation,* replaces the construction of the 2-D table by testing the stability of a finite number of 1-D polynomials of degree n_2 (or n_1) using a certain associated 1-D stability testing algorithm. The overall complexity of the resulting new procedure is $O(n^4)$ ($n = n_1 = n_2$). A more detailed count of operations and its comparison to other available solutions indicates that the new procedure has apparently an unprecedented low count of arithmetic operations.

The paper is organized as follows. The next section cites the tabular 2-D stability test from [14] and argues why it can be simplified into an $O(n^4)$ solution. Section III presents a new 1-D stability tests that may be used to sample this 2-D tabular test at desirable values along the unit circle. Section IV derives a simple solution of the required interpolation problem. The 2-D stability testing procedure that results from combining these components is presented in Section V. Section VI evaluates the computational cost of the procedure and compares it with other available solutions. Various comments, including a revealing comparison with an early numerical solution called the mapping method, conclude the paper.

II. Preliminaries

The problem considered in this paper is defined as follows.

Problem Statement: Given a 2-D (bivariate) polynomial

$$D(z_1, z_2) = [1, z_1, \ldots, z_1^{n_1}]D[1, z_2, \ldots, z_2^{n_2}]^t \qquad (1)$$

of degree (n_1, n_2), where $D = (d_{i,k})$, the matrix of coefficients, is a real or complex valued matrix, determine whether it does not vanish in the closed exterior of the unit bi-circle, viz.,

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

where $T = \{z: |z| = 1\}$, $V = \{z: |z| > 1\}$, $\overline{V} = V \cup T$.

A $D(z_1, z_2)$ that satisfies (2) will be called stable. (Other conventions for defining stability are also used in the literature on this topic. Bringing them to terms with the current notation may require some simple adjustment, e.g., reversion of the matrix

Manuscript received March 12, 1999; revised November 28, 2000. This paper was recommended by Associate Editor P. Rentrop.

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

Publisher Item Identifier S 1057-7122(01)05391-0.

D, cf. [1].) A 1-D polynomial $p(z)$ (with real or complex coefficients) such that

$$p(z) = [1, z, \ldots, z^n][p_0, \ldots, p_n]^t \neq 0 \qquad \forall z \in \overline{V} \tag{3}$$

will too be called stable.

The condition (2) is the key problem in testing stability of 2-D discrete systems. Its solution will not be restricted to assuming that D is real but simplifications that the real case admits will be specified. Presentation of a 1-D stable polynomials in a manner similar to the definition of 2-D stable polynomials, as above, is constructive for the solution of the stated problem. It however hides significant differences between stability of 1-D and 2-D systems. Most importantly, the fundamental theorem of algebra, according to which $p(z)$ has n zeros (and therefore it is stable if and only if they all reside inside the unit-circle), does not hold for polynomials with more than one variable. This deficiency complicates and limits the means available for handling stability as well as other issues in the design of multidimensional systems, [1]–[4].

This paper follows the convention and notation in [14]. Accordingly, polynomials and arrays may be interchanged, as demonstrated in (1). The more compact notation, $D(z_1, z_2) = \mathbf{z}_1^t D \mathbf{z}_2$, where $\mathbf{z} := [1, z, \ldots, z^i, \ldots]^t$ is a vector of length depending on context, may also be used. Similarly, a vector p is associated with a 1-D polynomials, $p(z) = \mathbf{z}^t p$ as in (3). The balanced polynomial is defined for a polynomial $p(z) = [1, s, \ldots, s^m]p$ by $p(\tilde{s}) = s^{-m/2}p(z) = \tilde{\mathbf{s}}^t p$, where $\tilde{\mathbf{s}} = [s^{-m/2}, \ldots, 1, \ldots, s^{m/2}]^t$ having an appropriate length. Reversion of matrices and vectors are defined and denoted by $D^{\sharp} := JD^{\star}J$ for a matrix and $e_k^{\sharp} := Je_k^{\star}$ for a vector, where J is the reversion matrix (a square matrix with 1's on its main anti-diagonal and zeros elsewhere), and $\star$ denotes complex conjugate. A vector and a polynomial are called symmetric if $p = p^{\sharp}$ or $p(z) = p^{\sharp}(z)$. A matrix with the property $D = D^{\sharp}$ and a 2-D polynomial such that $D(z_1, z_2) = D^{\sharp}(z_1, z_2)$ are called centro-symmetric. A matrix E_m has an index to indicate its position in the 2-D table. The columns of such a matrix carry this index in brackets, $E_m = [e_{[m]0}, e_{[m],1}, \ldots]$.

The tabular 2-D stability test in [14], consists of the following algorithm and theorem.

Algorithm 1: Obtain for $D(z_1, z_2)$ a sequence of centro-symmetric 2-D polynomials $\{E_m(s, z), m = -1, 0, \ldots, n = n_2\}$ as follows.

$$E_{-1}(s, z) = (z-1)(D(s, z) - D^{\sharp}(s, z)) \tag{4a}$$

$$E_0(s, z) = D(s, z) + D^{\sharp}(s, z), \qquad q_{-1}(s) = 1. \tag{4b}$$

For $m = 0, \ldots, n_2 - 1$ do

$$g_m(s) = e_{[m-1]0}(s)e_{[m]0}^{\sharp}(s) \tag{5a}$$

$$q_m(s) = e_{[m]0}(s)e_{[m]0}^{\sharp}(s) \tag{5b}$$

$$zE_{m+1}(s, z) = \frac{g_m(s)E_m(s, z) + g_m^{\sharp}(s)zE_m(s, z) - q_m(s)E_{m-1}(s, z)}{q_{m-1}(s)}. \tag{5c}$$

Theorem 1: $D(z_1, z_2)$ is stable if, and only if, the following three conditions hold.

1) $D(z, 1) \neq 0 \, \forall z \in \overline{V}$.
2) $D(1, z) \neq 0 \, \forall z \in \overline{V}$.
3)

$$\epsilon(s) := \frac{E_n(s, z)}{E_0(s, 1)} \neq 0 \qquad \forall s \in T \tag{6}$$

where $\epsilon(s)$ is a symmetric polynomial in s of degree $2n_1n_2$ that becomes available at the end of applying Algorithm 1 to $D(z_1, z_2)$.

The polynomial $E_{-1}(s, z)$ in Algorithm 1 has degree $(n_1, n_2 + 1)$, and the subsequent polynomials $E_m(s, z)$, $m = 0, 1, \ldots, n_2$, are of degrees $(\ell(m), n_2 - m)$, where $\ell(m) = (2m + 1)n_1$. The divisions by $q_{m-1}(s)$ are exact. Namely, the (symmetric) polynomial $q_{m-1}(s)$ divides the numerator polynomial with no remainder such that the result is indeed a 2-D polynomial. This division is responsible for attaining linear growth as function of m for the row sizes of the matrices E_m, instead of an otherwise exponential growth [14]. The polynomials $E_m(z_1, z_2)$ are centro-symmetric ($E_m = E_m^{\sharp}$). The variables (z_1, z_2) were replaced by (s, z) for convenience. The first variable will be most often regarded as taking values on the unit circle, $s \in T$. The polynomial $\epsilon(s)$ that celebrates in condition iii) is a symmetric 1-D polynomial obtained by exact division of $E_n(s, z)$ [of degree $(2n_1n_2 + n_1, 0)$] by $E_0(s, 1)$ (of degree n_1). It follows from its symmetry that $\epsilon(\tilde{s}) = s^{-n_1n_2}\epsilon(s)$ (its so called "balanced" form) is real on T. In fact, it has been shown in [14] that it is possible to replace condition iii) in Theorem 1 by the positivity condition

$$\epsilon(\tilde{s}) > 0 \qquad \forall s \in T. \tag{7}$$

The testing of the condition (6) can be carried out in $O(n^4)$ operations ($n = n_1 = n_2$). For example, the testing of this polynomial in the real case requires approximately $n_1^2n_2^2$ real multiplications using the method in [15]. However, the overall cost of this 2-D stability test is dominated by a higher cost involved in the construction of the table. A count of arithmetic operations can be carried out assisted by the following facts. The m-th recursion step consists of two convolutions and one deconvolution per column. $g_m(s)$ of degree $\ell(m-1) + \ell(m)$ multiplies $e_{[m]k}(s)$ of degree $\ell(m)$, $q_m(s)$ of degree $2\ell(m)$ multiplies $e_{[m-1]k}(s)$ of degree $\ell(m-1)$, and the resulting numerator polynomials of degrees $\ell(m-1) + 2\ell(m)$ are divided by $q_{m-1}(s)$ of degree $2\ell(m-1)$. Multiplication of two 1-D polynomials of degrees k_1 and k_2 requires $(k_1+1)(k_2+1)$ arithmetic operations. Dividing out a factor of degree k_1 from a polynomial of degree $k_1 + k_2$ (say $k_2 > k_1$) requires $(k_2 - k_1)(k_2 - k_1+1)/2$ arithmetic operations. The above counts can be halved by using the centro-symmetries toc calculate polynomials in s (the columns of the coefficient matrices) till only half of their full degree. Following this guide, an exact count for for the real and complex case can be obtained but the weary details will be skipped. It is enough to realize that the cost has $O(n_{1,2}^6)$ complexity, where $O(n_{1,2}^k)$" is used here and after to denote a polynomial expression with terms $n_1^{\alpha_1}n_2^{\alpha_2}$ such that $\alpha_1 + \alpha_2 \leq k$.

This paper aims to obtain from this tabular test, a solution of $O(n^4)$ overall complexity (returning for convenience to the $n = n_1 = n_2$ assumption) anticipated according to the following observations. The above overall $O(n^6)$ cost is caused by the computation required for the construction of the 2-D table. However, according to Theorem 1, the construction of the 2-D table is required only in order to get $\epsilon(s)$ at its end. It is well known that a polynomial of degree N can be determined from its value at $N+1$ distinct points. It is also known that this interpolation problem, that amounts to solving a Vandermonde set of equations, can be solved in $O(N^2)$ complexity, (see for example [16]). Consequently, (with $N = n^2$), $\epsilon(s)$ can be obtained from a finite set of $O(n^2)$ known value in $O(n^4)$ operations. Since then, as already said, the condition (6) can be tested in $O(n^4)$ operation, an overall $O(n^4)$ solution is possible if sample values of $\epsilon(s)$ can be obtained in $O(n^2)$ operations per value. The next section will show an efficient $O(n^2)$ algorithm to obtain sample values of $\epsilon(s)$. It subsequent section will then bring a simple and direct formula to recover $\epsilon(s)$ from these sample values.

III. Companion 1-D Stability Test

This section begins by singling out an algorithm that can be used to obtain sample values of $\epsilon(s)$. Afterwards, the new algorithm is turned into a 1-D stability test by posing on it necessary and sufficient conditions for 1-D stability.

The sought algorithm is obtained basically by reverting the manner used to derive the tabular test in [14] from the 1-D stability test of [17]. It is possible to regard Algorithm 1 as a recursion of 1-D polynomial in the variable z with coefficients that are polynomials in s. The degree in s of $E_m(s, z)$ is $\ell(m) = (2m+1)n_1$. Multiplying the two sides of the recursion (5c) by $s^{-\ell(m+1)/2}$ and breaking this factor properly among polynomials of s in the right hand side leads to an equivalent algorithm (in the sense of propagating the same arrays) that amounts to replacing everywhere in Algorithm 1 s by $\tilde{s}$, i.e., $E_m(s, z)$ by $E_m(\tilde{s}, z) = \tilde{\mathbf{s}}^t E_m \mathbf{z}$, $g_m(s)$ by $g(\tilde{s}) = \tilde{\mathbf{s}} g$ and $q_m(s)$ by $q(\tilde{s}) = \tilde{\mathbf{s}} g$. A balanced polynomial satisfies $p^{\sharp}(\tilde{s}) = [p(\tilde{s})]^\star$ for values $s \in T$. Thus, for $s \in T$, reversion is implemented by conjugation. Presenting Algorithm 1 by the balanced polynomials helps to realize that, for any fixed value $\tilde{s}_o \in T$, the action of Algorithm 1 on $D(\tilde{s}_0, z)$ corresponds to applying to $p(z) = D(\tilde{s}_o, z)$ the next algorithm.

Algorithm 2: Consider the polynomial

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

where p_k are complex scalars. Obtain a sequence of polynomials $\{e_m(z) = \sum_{k=0}^{n-m} e_{m,k} z^k, m = -1, 0, 1, \ldots, n\}$ and scalars $\{\sigma_m, m = -1, 0, 1, \ldots, n\}$ as follows:

$$e_{-1}(z) = (z-1)(p(z) - p^{\sharp}(z)) \tag{9a}$$

$$e_0(z) = p(z) + p^{\sharp}(z), \qquad q_{-1} = 1 \tag{9b}$$

$$\sigma_{-1} = 0, \quad \sigma_0 = 1 \tag{9c}$$

For $m = 0, \ldots, n-1$ do

$$g_m = e_{m-1,0} e_{m,0}^\star, \qquad q_m = e_{m,0} e_{m,0}^\star \tag{10a}$$

$$z e_{m+1}(z) = \frac{(g_m + g_m^\star z) e_m(z) - q_m e_{m-1}(z)}{q_{m-1}} \tag{10b}$$

$$\sigma_{m+1} = \frac{(g_m + g_m^\star)\sigma_m - q_m \sigma_{m-1}}{q_{m-1}}. \tag{10c}$$

Algorithm 2 may be regarded as the projection of Algorithm 1 (with balanced polynomials in the first variable) on $D(\tilde{s}_o, z)$ for a fixed $s_o \in T$ obtained through the substitutions, $E_m(\tilde{s}, z) \to e_m(z)$, $g_m(\tilde{s}) \to g_m = e_{m-1,0} e_{m,0}^\star$, and $q_m(\tilde{s}) \to q_m = e_{m,0} e_{m,0}^\star$. The assumption on $p(z)$ in (8) will hold in its following application.

The comparison with Algorithm 1 reveals that a parallel and separate recursion that produces a sequence of scalars, $\{\sigma_m\}$, has been added to Algorithm 2. These scalars play an important role in the next two theorems. Setting $z = 1$ in Algorithm 1 reveals that the σ_m's correspond, for a fixed $s \in T$, to

$$\frac{E_m(\tilde{s}, 1)}{E_0(\tilde{s}, 1)} \to \sigma_m.$$

It also confirms the form of the recursion (9c) and its initiation (8c). Furthermore, setting $z = 1$ into Algorithm 2 shows that normally the recursion (8c) provides an alternative way to obtain the values $\sigma_m = e_m(1)/e_0(1)$ that requires less computation then summing the coefficients of $e_m(z)$. The more important reason for producing σ_m's with a separate recursion is for cases when $\mathcal{R}e\{p(1)\} = 0$ implies $e_m(1) = 0$ for all m. In such cases the separate recursion for the σ_m's circumvents the otherwise ambiguous 0/0 expression.

Theorem 2: Assume Algorithm 2 is applied to $p(z)$ (8). Then $p(z)$ is stable if and only if

$$\sigma_m > 0, \qquad m = 1, \ldots, n. \tag{11}$$

Furthermore, if an $e_{m,0} = 0$ occurs then $p(z)$ is not stable.

The combination of Algorithm 2 with Theorem 2 forms a new stability test for 1-D polynomials in its own right. It will be currently referred as the "companion 1-D stability test." One way to prove Theorem 2 is through relations between respective polynomials here and in [17]. In the current context, Theorem 2 can be more readily deduced from the proof brought for Theorem 1 in [14]. One delicate point that may need attention concerns a $p(z)$ for which $\mathcal{R}e\{p(1)\} = 0$. For this situation, it is reminded that Algorithm 2 may be regarded as the effect of Algorithm 1 on $D(\tilde{s}, z)$ at any fixed $s \in T$. However, in the context of Algorithm 1, $\sigma_m(s) := E_m(s, 1)/E_0(s, 1)$ are polynomials [14] and therefore continuous for $s \in T$ in the vicinity of a $s_o \in T$ for which $E_0(s_o, 1) = 0$. In the course of the proof for Theorem 1 in [14] it was shown that stability of 1-D polynomials $p_s(z) = D(\tilde{s}, z)$ for s in the vicinity of such a $s_o \in T$ is not obstructed in the limit $s \to s_o$. It follows that the recursion (9c), that circumvents the 0/0 in such situations, produces as $s \to s_o$ the same value that Algorithm 1 assigns to the polynomial $E_m(\tilde{s}, 1)/E_0(\tilde{s}, 1)$ at $s = s_o$.

The next theorem summarizes how Algorithm 2 may be used to obtain sample values of $\epsilon(\tilde{s})$ at desirable $s \in T$ without construction of the 2-D stability table.

Theorem 3: Assume Algorithm 1 assigns to $D(z_1, z_2)$ the polynomial $\epsilon(s) = \mathbf{s}^t \epsilon$ defined in (6) and let $\epsilon(\tilde{s}) = \tilde{\mathbf{s}}^t \epsilon$ denote the corresponding balanced polynomial. For a fixed $s_m \in T$ denote $b_m := \epsilon(\tilde{s}_m)$ and define the 1-D polynomial of degree n_2 $p_{s_m}(z) = D(\tilde{s}_m, z)$. Apply to $p_{s_m}(z)$ the companion 1-D stability test. If $p_{s_m}(z)$ is determined as not stable then $D(z_1, z_2)$ is not stable. If $p_{s_m}(z)$ is determined as stable then Algorithm 1 produces $\sigma_{n_2} = b_m$ at its end.

The validity of this theorem has been clarified in the course of establishing Theorem 2. Algorithm 2 may provide sample values for $\epsilon(\tilde{s})$ also when $p_{s_m}(z) = D(\tilde{s}_m, z)$ is not stable. However, if $p_{s_m}(z)$ is not stable then $D(z_1, z_2)$ is not stable and its testing may already be terminated. Note also that a $p_{s_m}(z)$ [hence $D(z_1, z_2)$] can be declared as not stable as soon as a necessary condition for stability, according to Theorem 2, is found not to hold.

Remark 1: The new testing procedure will use Algorithm 2 repeatedly, therefore its cost of computation affects noticeably the overall complexity of the 2-D stability test. Algorithm 2 may be carried out in $1.5n^2 + O(n)$ real multiplications and $2n^2 + O(n)$ additions for a complex polynomial of degree n by regarding it as a recursion with just two multipliers g_m/q_{m-1} and q_m/q_{m-1}. It is possible to obtain a sample value b_m in even $n^2 + O(n)$ multiplications by using the original recursion form of [17] that has only one multiplier per recursion step. It has then to be used with a correcting post-multiplying factor that requires only $O(n)$ operations. This scheme achieves better "squeezing" of the final overall cost of computation but it does so at the expense of obtaining the b_m in a less desirable manner.

Remark 2: The author has found the study of 2-D stability and 1-D stability to keep fertilizing each other. The tabular stability test in [14] stems from the modified 1-D stability test [17]. Considerations made there to improve the efficiency of the 2-D tabular are seen now to lead to a new 1-D stability test that is different from the test in [17] in several interesting ways. One difference is in the initial requirement posed on $p(z)$ in the two tests. The requirement in [17] was that $\mathcal{R}e\{p(1)\} \neq 0$ while here it is relaxed to $p(1) \neq 0$. The former assumption poses no difficulty on a stand-alone test [if $p(1) \neq 0$ and $\mathcal{R}e\{p(1)\} = 0$ then $jp(z)$ can be tested instead]. However, exception for an $s_o \in T$ for which $p(z) = D(\tilde{s}_o, z)$ is such that $\mathcal{R}e\{p(1)\} = 0$ is currently not tolerated. The relaxed condition on $p(1)$ admits a uniform treatment of all 1-D polynomials that is crucial in the current application. It is possible to relax the assumption on $p(z)$ also in [17] to just $p(1) \neq 0$ by adding there too a separate recursion for "σ_m" scalars to circumvent ambiguous sign variation rule in cases when $e_m(1) = 0$ for all m.

IV. The Interpolation Problem

This section brings a simple way to determine the vector of coefficients $\epsilon = [\epsilon_0, \ldots, \epsilon_{2M}]^t$, $M := n_1 n_2$, of $\epsilon(s) = \mathbf{s}^t \epsilon$ from known values of $\epsilon(\tilde{s}_m) = \tilde{\mathbf{s}}^t \epsilon$ at a set of samples $s \in T$. [Required values of $\epsilon(\tilde{s})$ at point $s_m \in T$ will be provided by the companion 1-D stability test as described in Theorem 3.] This interpolation problem is known to be solvable in $O(N^2)$ for a polynomial of degree N (e.g., the algorithm in [16], that works also for complex polynomials, may be used). However, a more efficient solution can be obtained as shown below paying attention to the specifics of the current interpolation problem. A solution of reduced complexity is attainable by minimizing the number of required sample values (in order to minimize the number of times Algorithm 2 is applied) and by exploiting the fact that the sample values are on T.

Since the polynomial $\epsilon(s) = \mathbf{s}^t \epsilon$ has degree $2M$, it can be determined from knowing its value at $2M + 1$ distinct points. Therefore, the balanced polynomial, $\epsilon(\tilde{s}) = \mathbf{s}^{-M} \epsilon(s)$ can too be determined from values $b_m = \epsilon(\tilde{s}_m)$ at $2M + 1$ distinct points $s_m \in T$. A collection of $2M + 1$ values of $\epsilon(\tilde{s}_m)$ at distinct points produces the next set of equations

$$[s_m^{-M}, s_m^{-M+1}, \ldots, s_m^{-1}, 1, s_m, \ldots, s_m^{M-1}, s_m^M]\epsilon = b_m, \quad m = 0, 1, \ldots, 2M. \tag{12}$$

This set has to be solved for $\epsilon := [\epsilon_0, \ldots, \epsilon_{2M}]^t$. For the case of a real D, it is of advantage, as will become apparent in a moment, to choose $2M$ interpolation points in conjugate pairs. Choosing the points equally spaced along T acquires the process with a discrete Fourier transform (DFT)-like orthogonality property that simplifies the solution of the set of equations. An adequate choice of interpolation points that satisfies both requirements is given by

$$\theta := \frac{2\pi}{2M+1} \quad w := e^{j\theta} \quad s_m := w^{-M+m}, \quad m = 0, 1, \ldots, 2M \tag{13}$$

($j = \sqrt{-1}$). For this choice, the set of equations (12) becomes $Q\epsilon = b$ where $b = [b_0, \ldots, b_{2M}]^t$ is the vector of known sample values, and Q has the form as in the matrix shown at the bottom of the next page. The matrix Q is symmetric, $Q^t = Q$, as well as centro-symmetric, $JQJ = Q$. The columns of $Q = [v_{-M}, \ldots, v_{-1}, v_0, v_1, \ldots, v_M]$ are given by the vectors

$$v_k := [w^{-Mk}, w^{-(M-1)k}, \ldots, w^{-k}, 1, w^k, \ldots, \cdot w^{(M-1)k} w^{Mk}]^t, \qquad k = 0, \pm 1, \ldots, \pm M.$$

The inner product of two vectors in this set reveals the next orthogonality

$$v_{-i}^t v_k = w^{-M(k-i)} \frac{1 - w^{(2M+1)(k-i)}}{1 - w^{k-i}} = \begin{cases} 2M+1, & k = i \\ 0, & k \neq i. \end{cases}$$

It follows that

$$Q^{-1} = \frac{1}{2M+1} QJ.$$

Therefore, an explicit solution to $Q\epsilon = b$ is given by

$$\epsilon = \frac{1}{2M+1} QJb.$$

Using the symmetry $\epsilon = J\epsilon^\star$, it suffices to read only the upper half rows of the this solution. The result is the expression

$$\begin{aligned}\epsilon_{M-m} =& \frac{1}{2M+1}\Bigg\{b_M + \sum_{k=1}^{M-1}[(b_{M+k}+b_{M-k})\cos(mk\theta) \\ &+ j(b_{M+k}-b_{M-k})\sin(mk\theta)]\Bigg\}, \\ & m=0,\ldots,M \\ \epsilon_{M+m} =& \epsilon^\star_{M-m}, \qquad m=1,\ldots,M. \end{aligned} \tag{14}$$

If, in addition D is real, then ϵ is real. Therefore the value b_m of $\epsilon(\tilde{s})$ at s_m is equal to the value b_{2M-m} at $s_{2M-m} = s_m^{-1}$. The resulting relations, $b_{M+k} = b_{M-k}$, $m = 0, \ldots, M-1$ simplifies (14) to

$$\begin{aligned}\epsilon_{M-m} =& \frac{1}{2M+1}\left\{b_M + 2\sum_{k=1}^{M} b_{M-k}\cos(mk\theta)\right\}, \\ & m=0,\ldots,M \\ \epsilon_{M+m} =& \epsilon_{M-m} \quad m=1,\ldots,M. \end{aligned} \tag{15}$$

V. The New 2-D Stability Test

The proposed method for testing whether the 2-D polynomial $D(z_1, z_2)$ of (1) satisfies condition (2) is based on Theorem 1 and the components prepared in the previous sections to obtain $\epsilon(s)$ more efficiently. It is summarized in the next procedure. In the following, "exit" is used to mark points which, if reached allow early termination of the procedure with the conclusion "$D(z_1, z_2)$ is not stable."

A Procedure for Testing $D(z_1, z_2)$:

Step 1. Determine whether $D(z, 1)$ is 1-D stable. If not stable—"exit."

Step 2. Let $M = n_1 n_2$, $\theta = 2\pi/(2M+1)$, $w = e^{j\theta}$ ($j = \sqrt{-1}$). For $m = 0, 1, \ldots, M$ for a real D (and also for $m = M+1, \ldots, 2M$ for a complex D) do:

Set $s_m = w^{-M+m}$. Apply the companion 1-D stability (Algorithm 2 + Theorem 2) to $p_{s_m}(z) = D(\tilde{s}_m, z)$. If $p_{s_m}(z)$ is not 1-D stable (as soon as a $e_{[i]0} = 0$, $-1 \le i \le n_2$ or $\sigma_i \le 0$, $1 \le i \le n_2$ is observed)—"exit." Otherwise, retain $\sigma_{n_2} (>0)$ as $b_m := \sigma_{n_2}$.

Step 3. Obtain the polynomial $\epsilon(s) = \sum_{i=0}^{2M+1} \epsilon_m s^i$ from the values $b_m (>0)$, using (15) for a real D [or (14) for a complex D].

Step 4. Examine the condition "$\epsilon(s) \neq 0 \ \forall s \in T$." $D(z_1, z_2)$ is stable if and only if this condition is true and the current step has been reached without an earlier "exit."

Step 1 corresponds to condition i) of Theorem 1. Condition 2) of this theorem is examined in step 2 at $m = 0$. Step 2 obtains sample values using the companion 1-D stability test and Theorems 2 and 3. Note that once the test passed step 1, all 1-D stability tests in step 2 satisfy the assumption that $p_{s_m}(1) \neq 0$ required for Algorithm 2. Step 3 constructs $\epsilon(s)$ using the formulas developed in Section III. Finally, step 4 implements condition iii) of Theorem 1. Clearly, it is always possible to test $\mathbf{z_1}^t D^t \mathbf{z_2}$ instead of $\mathbf{z_1}^t D \mathbf{z_2}$. It will become apparent from the count of arithmetic operations in the next section, that to reduce computation, it is better to use a matrix with less columns than rows. Namely, if in (1) $n_1 < n_2$, it is preferable to apply the procedure to $\mathbf{z_1}^t D^t \mathbf{z_2}$.

VI. Evaluation

An approximate count of arithmetic operations for the proposed procedure will now be carried out. Counts are in terms of real multiplications and additions. Multiplication of two complex numbers are counted as four real multiplications and two real additions, multiplication of a real number by a complex number as two real multiplications, and the symmetry of the polynomials is used to compute only half of the coefficients. The count is approximate in that it retains only the leading terms in polynomial expressions of the precise count. The count will also assume that the last step is performed by the methods in [15] and [18] or [17], which are the procedures of least count of operations available to determine zero location of a 1-D polynomial with respect to the unit circle.

Step 1 is a 1-D stability test for a real polynomial of degree $n = n_1$. It can be carried out in $0.25n^2 + O(n)$ multiplications and $0.5(n^2) + O(n)$ additions for a real polynomial and by $n^2 + O(n)$ multiplications and $2n^2 + O(n)$ additions for a complex polynomial using [15] and [18] or [17]. These $O(n^2)$ counts are negligible compared to the overall $O(n^4)$ complexity.

Step 2 involves in the real case $n_1 n_2$ companion stability tests of complex 1-D polynomials [and one real polynomial—$p_{s_0}(s)$] each of degree n_2. The number of complex 1-D tests that is

$$Q = \begin{bmatrix} w^{MM} & w^{M(M-1)} & \ldots & w^{M} & 1 & w^{-M} & \ldots & w^{-(M-1)M} & w^{-MM} \\ w^{(M-1)M} & w^{(M-1)(M-1)} & \ldots & w^{M-1} & 1 & w^{-(M-1)} & \ldots & w^{-(M-1)(M-1)} & w^{-(M-1)M} \\ \vdots & & & & \vdots & & & & \vdots \\ 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & 1 \\ \vdots & & & & \vdots & & & & \vdots \\ w^{-(M-1)M} & w^{-(M-1)(M-1)} & \ldots & w^{-(M-1)} & 1 & w^{M-1} & \ldots & w^{(M-1)(M-1)} & w^{(M-1)M} \\ w^{-MM} & w^{-M(M-1)} & \ldots & w^{-M} & 1 & w^{M} & \ldots & w^{(M-1)M} & w^{MM} \end{bmatrix}$$

required doubles when D is complex valued. Algorithm 2 can test a complex polynomial of degree n in $1.5n^2 + O(n)$ real multiplications, and $2n^2 + O(n)$ real additions. Thus, step 2 requires in the real case $1.5n_1n_2^3$ real multiplications and $2n_1n_2^3$ real additions + $O(n_{1,2}^3)$ (in principle, as pointed in Remark 1 the count of multiplication can be squeezed even further). The counts double for a complex D.

Step 3 requires $n_1^2n_2^2$ real multiplications and additions for a real D. The count doubles for a complex D.

Step 4 may be tested by the zero location tests of [15] or [17] in $n_1^2n_2^2$ multiplications and $2n_1^2n_2^2$ additions $+O(n_{1,2}^3)$ when D and therefore $\epsilon(s)$ is real. If D is complex then $\epsilon(s)$ is complex and step 4 require $4n_1^2n_2^2$ multiplications and $8n_1^2n_2^2$ additions $+O(n_{1,2}^3)$ using the tests in [18] or [17].

Summing the counts for the four steps yields the overall cost of computation for the procedure. The procedure requires $2n_1^2n_2^2 + 1.5n_1n_2^3$ real multiplications and $3n_1^2n_2^2 + 2n_1n_2^3$ real additions $+O(n_{1,2}^3)$ for a real D and it requires $6n_1^2n_2^2 + 3n_1n_2^3$ real multiplications and $10n_1^2n_2^2 + 4n_1n_2^3$ real additions $+O(n_{1,2}^3)$ for a complex D.

The following account on the computational requirement of previous solutions will use $n_1 = n_2 = n$ for simplicity. All references cited below considered a real 2-D polynomial a case for which the current procedure requires $3.5n^4$ (or even just $3n^4$) multiplications and $5n^4$ additions $+O(n^3)$. The author is not aware of previous solutions by other authors for the case of complex 2-D polynomials. [The immittance $O(n^6)$ tabular tests in [14], [20] were proposed for also the complex case.]

The methods for solving this problem algebraically (i.e., in finite number of operations) may be classified into tabular and determinant methods [1], [4]. The first tabular test of Maria and Fahmy [19] can be shown to be of $O(n^2 4^n)$ complexity. This can be shown to be also the complexity of all tabular tests proposed in the following two decades till and including [8]. (A tabular test is bound to have this order of complexity if it ends with a polynomial of degree $n2^{n-1}$ or a symmetric polynomial of double degree). The determinant methods were based on testing determinants of various "stability" matrices (Schur–Cohn Bezuotian, Sylvester resultants, Inner matrices and more) with polynomial entries, e.g., [7] and earlier works surveyed in [4]. These determinant solutions are too of exponential complexity. O'Connor and Huang commented in [1] that a solution of $O(n^5)$ must exist for the problem (though not necessarily in a simple to manage form) because they were aware of the existence of an $O(n^5)$ algorithm to determine the determinant of a matrix of size $n \times n$ whose entries are polynomials of degree up to n. However, only recently have tabular tests stepped down from exponential complexity to polynomial complexity. As shown here already, the complexity of the tabular test [13], [14] has $O(n^6)$ complexity. This can be shown to be also the complexity of the test by Hu and Jury in [10] and an alternative immittance-type tabular test in [12], [20]. Gu and Lee proposed in [6] to interpolate the determinant of the Schur-Cohn matrix using Cholesky factorization pointing on the availability of efficient software packages for the required factorization. However, they do not detail a specific procedure and do not provide a count of operations. Kurosawa, Yamada and Yokokawa proposed in [11] a similar method that obtains the determinant of the polynomial matrix using DFT and has $O(n^5)$ complexity. Barret and Benidir suggested in [9] a solution that uses a generalized Levinson algorithm to interpolate the resultant matrix and showed that it requires approximately $23n^4 + O(n^3)$ real multiplications and $23.5n^4 + O(n^3)$ real additions. The author is not aware of any other solution of lower complexity for the real case and of no other solutions at all for the complex case by other authors. Therefore, the procedure presented here is considered to be the solution of least count of operations available at this time for testing stability of 2-D polynomial with real or complex coefficients.

It has been rightly said already in [1] that 2-D stability tests should be evaluated in terms of algorithmic and computational complexity and in terms of accuracy and numerical effects. The above comparison indicates that the current procedure excels in computational complexity. The programming of the current procedure is also quite simple—it involves the repeated use of a single routine—algorithm 2 (this algorithm can also be extended to solve the zero location problem of step 4). The significant lower computational complexity attained by telepolation and exploiting the symmetries in the current procedure is expected to improve numerical accuracy of the procedure compared to other 2-D stability tests of higher complexity. An alternative to the procedure presented here is possible by applying telepolation to the immittance 2-D tabular in [12], [20]. The resulting algorithm, that has been described in [21], has a comparable count of operations. The two procedures differ in fine details that may affect their relative numerical accuracy or other merits that have not yet been investigated. An experimental study of the numerical accuracy of several earlier 2-D stability tests that has been carried out in [22], as well as our less exhaustive numerical experience so far with the method proposed here, support expectation for improved numerical accuracy compared to more complex algorithms. The relative merits of the emerging new solutions and the developing of numerically robust stability tests remain subjects for further study.

VII. Concluding Remarks

A new procedure for testing the stability of 2-D discrete-time systems has been developed. The procedure determines whether a two-variable polynomial has no zeros in the closed exterior in apparently unprecedented low count of operations. The new 2-D stability test profits on the efficiency of the immittance-type tabular test of [13], [14] (that has lower complexity than earlier tabular tests) and its simple stability conditions (testing only one last polynomial for no zeros on the unit circle). It accomplishes further reduction in complexity by realizing that the burden of computation of the tabular test is dominated by the cost of computation of the table and then shows that further significant reduction in computation is attainable by using a collection of (properly designed) 1-D stability tests of low degree to bring forth the last entry of the table without its full construction.

The new approach, called telepolation contributes to the theory of multidimensional stability an important observation that was not known for it before. It shows that testing the stability of a two-dimensional system polynomial can be carried out by a well defined finite number of 1-D stability tests of a common specific form. Since the major gap in the mathematics for m-D ($m \geq 2$) systems occurs upon moving from 1-D to 2-D systems, the extension of the approach demonstrated here for the 2-D case to stability testing of higher dimensional systems poses no conceptual obstacles.

It is interesting to compare the current approach with one of the earliest solutions to the problem called the mapping method. It is based on a simplification proposed first by Huang [1] that admits replacement of the condition (2) by the condition $D(s, z) \neq 0 \, \forall s \in T$ and $z \in \overline{V}$ (plus a 1-D stability). This simplification, that is cited and was used also in [14], has become the starting point of virtually all 2-D stability tests. The mapping method, proposed by O'Connor and Huang [1, Sec. 4.10], suggested to test the above condition by considering a set of 1-D polynomials obtained by sampling $D(s, z)$ at grid of values $s \in T$ and then test their stability by finding their roots (called then the root mapping method) or by any algebraic 1-D stability test. The mapping method is a numerical method that does not carry a cost tag of finite number of arithmetic operations. It provides an approximate answer whose reliability increases with increasing the number of 1-D stability tests (the density of the grid). The similarity of the mapping and the telepolation methods is in that they both examine the stability of 1-D polynomials obtained by sampling along the unit circle one of the variables of $D(z_1, z_2)$. The contrast between the two solutions highlights the innovation in the telepolation approach from one more perspective. It shows that 2-D stability can be determined *exactly* and by using just a *finite* number of sample 1-D polynomials, if only they are obtained and tested by a qualifying 1-D stability tests.

The solution proposed in this paper is apparently the method of least count of operations for 2-D stability testing of discrete-time system polynomials known today. The fact that the procedure consists of a collection of 1-D stability tests makes its programming very simple. It also makes the procedure pass a path dense with necessary conditions for stability that is useful to reduce even further the computational effort wasted on an unstable 2-D polynomial.

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, Germany: Springer-Verlag, 1981, ch. 4.

[2] N. K. Bose, *Applied Multidimensional System Theory*. New York: 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] ——, "Modified stability table for 2-D digital filter," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 116–119, Jan. 1988.

[6] G. Gu and E. B. Lee, "A Numerical algorithm for stability testing of 2-D recursive digital filter," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 135–138, Jan. 1990.

[7] A. Kanellakis, S. Tzafestas, and N. Theodoru, "Stability tests for 2-D systems using the Schwarz form and the inners determinants," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 1071–1077, Sept. 1991.

[8] X. Hu and H. Yee, "Polynomial array for $F(z_1, z_2)$ on $|z_1| = 1$ and 2-D filter stability tests," *IEEE Trans. Signal Processing*, vol. 40, pp. 1579–1581, June 1992.

[9] M. Barret and M. Benidir, "A new algorithm to test stability of 2-D digital recursive filters," *Signal Process.*, vol. 37, pp. 255–264, 1994.

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

[11] K. Kurosawa, I. Yamada, and T. Yokokawa, "A fast 2-D stability test procedure based on FFT and its computation complexity," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 676–978, Oct. 1995.

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

[13] ——, "Stability testing of 2-D digital system polynomials using a modified unit circle test for 1-D complex polynomials," in *Proc. 29th Asilomar Conf. Signals, Systems & Computers*, 1995, pp. 617–621.

[14] ——, "Stability testing of 2-D discrete linear system polynomials by a 2-D tabular form," *IEEE Trans. Circuits Syst. I*, vol. 46, pp. 666–676, June 1999.

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

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The John Hopkins Univ. Press, 1983.

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

[18] ——, "A circular stability test for general polynomials," *Syst. Control Lett.*, vol. 7, no. 2, pp. 89–97, 1986.

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

[20] Y. Bistritz, "Immittance-type tabular stability test for 2-D LSI Systems based on a zero location test for 1-D complex polynomials," *Circuits Syst. Signal Process.*, vol. 19, no. 3, pp. 245–265, 2000.

[21] ——, "A stability test of reduced complexity for 2-D digital system polynomials," in *Proc. IEEE Int. Symp. Circuits Systems*, CA, May 1998.

[22] M. Barret and M. Benidir, "Behavior of stability tests for two-dimensional digital recursive filters when faced with rounding errors," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 319–323, 1997.

Yuval Bistritz (S'79–M'79–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 1977 and held an industrial research position from 1977 to 1979. From 1979 to 1984, he held various assistant and teaching position in the Department of Electrical Engineering, Tel Aviv University, and in 1987, he joined the department as a Faculty Member. From 1984 to 1986 he was a research scholar in the Information System Laboratory, Stanford University, Stanford, CA, working on fast signal processing algorithms. From 1986 to 1987 he was with AT&T Bell Laboratories, Murray Hill, NJ, and from 1994 to 1996 with the DSP Group, Santa Clara, CA, working as consultant on speech processing algorithms. His research interests are in the area of digital signal processing and system theory.