# ON JURY'S TEST FOR 2-D STABILITY OF DISCRETE-TIME SYSTEMS AND ITS SIMPLIFICATION BY TELEPOLATION

*Yuval Bistritz*

Department of Electrical Engineering
Tel Aviv University, Tel Aviv 69978, Israel

## ABSTRACT

The paper revises and simplifies Jury's tabular stability test for two-dimensional (2-D) discrete-time systems. The tabular test builds for a 2-D polynomial of degree $(n_1, n_2)$ a '2-D table' - a sequence of $n_2$ matrices or equivalently 2-D polynomials and then examines its last entry - a 1-D polynomial of degree $2n_1n_2$ for no zeros on the unit circle. Analysis of the cost of computation for the test is performed and shows that it is of $O(n^6)$ $(n_1 = n_2 = n)$, compared to previous tabular tests of exponential complexity. Next, we propose a new test based on telepolation - *tele*scoping the last entry of this 2-D table by inter*polation*. The table's construction is replaced by $n_1n_2 + 1$ stability tests of 1-D polynomials of degree $n_1$ or $n_2$. The resulting new 2-D stability test is shown to require a low $O(n^4)$ count of operations.

## 1. INTRODUCTION

Stability of two-dimensional (2-D) linear discrete-time (linear shift invariant) systems arises in many applications. The problem has been reviewed in several texts including [1] [2]. The key to stability determination of 2-D discrete systems is an efficient solution for the next problem.

**Problem statement**. Given a two-dimensional (2-D, bivariate) polynomial

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

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

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

Here, $T = \{z : |z| = 1\}$ , $U = \{z : |z| < 1\}$ , $V = \{z : |z| > 1\}$ are used to denote the unit circle, its interior, and its exterior, respectively, and the bar denotes closure, $\bar{V} = V \cup T$. (Some other notations are also used in the literature and may require a reversion of the coefficient matrix to conform with the present convention [3].)

A 2-D polynomial $D(z_1, z_2)$ that satisfies (2) will be called *stable*. Similarly a 1-D system polynomial $p(z)$ such that

$$p(z) = [1, z, \dots, z^n]p \neq 0 \qquad \forall z \in \bar{V}\,. \qquad (3)$$

will too be called stable. The above arranged similarity in the definition of 1-D and 2-D stable polynomials is useful for the solution of the problem but at the same time it hides some theoretical and practical difficulties in testing stability of higher dimensional systems that are not existent in the 1-D case [1] [2].

The current paper focuses on the 2-D stability test proposed in [4] [5]. This 2-D stability test evolves from Jury's so called *modified* 1-D stability test proposed by him in several versions in the last four decades. In a classification of the Schur-Cohn Marden-Jury (SCMJ) class of 1-D stability tests into four types these modified test fall into the C-type category [6]. The special property of C-type tests is that they associate the tested 1-D polynomial with stability tables that contain at specific locations entries identical with the principal minors of its Schur-Cohn Bezoutian matrix. This property has allowed [4][5] to adopt Siljak's simplification for determining positive definiteness of a polynomial matrix [7] and obtain 2-D stability tests with just a single so called 'positivity test'.

We first provide a concise and simple setting for Jury's 2-D stability test. Our version consists of an algorithm to construct the 2-D table and accompanying stability theorem that answer the stated problem. It differs in notation and in structure in that it uses our convention in other recent works on 2-D stability (cf. [3]), and in that it follows the basic form for C-type 1-D tests in [6] that is not identical with any of the previous modified Jury's C-type 1-D tests. (Our basic test form carries the principal minors of the Schur-Cohn matrix as the leading coefficients of the associated sequence of polynomials. For precise relations with all other published versions, see [6]).

We carry out for the first time an approximate count of operations for this 2-D stability test. It shows that the test is of $O(n^6)$ complexity (for $n = n_1 = n_2$) - a definite advantage over previous 2-D tabular stability tests of exponential complexity. Examination of the accompanying stability conditions reveal that the table is constructed merely to reach its last entry - presented by a symmetric polynomial of degree $2n_1n_2$ that has then to be tested for no zeros on the unit-circle. This last task can be carried out in $O(n^4)$ (again for $n_1 = n_2 = n$) operations. However, the tabular test has an overall $O(n^6)$ complexity because it is dominated by the cost of the table's construction.

Subsequently, the paper proceeds to its main new contribution - it obtains a new 2-D stability test whose overall cost of computation is $O(n^4)$ operations. This improvement is achieved by circumventing the table's construction and replacing it by a finite number of low degree 1-D stability tests from which the last entry of the table can be recovered by an efficient closed form interpolation formula. We call the proposed approach *telepolation*, standing for *tele*scoping

appeared in ISCAS 2000

by *inter*polation (the last entry of the sequence). A procedure to carry out the new 2-D stability test is detailed and its cost of computation and additional merits are evaluated.

## 2. THE 2-D AND ITS 1-D COMPANION TABULAR STABILITY TEST

### 2.1. notation

The notation that we use here has been detailed before on several occasions, e.g. in [3]. Briefly, it admits interchangeable use of polynomial and array notation. For example, a matrix $D$ may be associated with a 2-D polynomial by $D(z_1, z_2) = \mathbf{z_1}^t D \mathbf{z_2}$ as in (1). The vector $\mathbf{z} := [1, z, \ldots, z^i, \ldots]^t$ is of length depending on context. This notations is equally used to associate vectors with a 1-D polynomials, e.g. $p$ with $p(z) = \mathbf{z}^t p$. The letter $\mathbf{s}$ is reserved for $s \in T$. $D^\sharp := JD^\star J$ and $e_k^\sharp := Je_k^\star$ denote matrix and vector (conjugate) reversion, where $J$ is the reversion matrix (a matrix with $1'$s on the main anti-diagonal and $0'$s elsewhere), and $\star$ denotes complex conjugate. The notation can be used to set the 2-D tabular test algorithm in also a matricial notation that is more transparent for programing by a vector oriented language like Matlab. For brevity we shall adhere here mostly to the polynomial pan of the notation which is also the more instrumental presentation for the derivation and of the new results.

### 2.2. The 2-D tabular test

Our formulation for the 2-D stability test of [5] consists of the next Algorithm 1 and Theorem 1.

**Algorithm 1.** A 2-D table that consists of a sequence of matrices $\{C_m,\ m = n-1, ,\ldots 0\}$, or equivalently, a sequence of 2-D polynomials $\{C_m(s, z) = \sum_{k=0}^m c_{[m]k}(s) z^k,\ m = n-1, \ldots, 0\}$ is assigned to the tested $D(z_1, z_2)$ (1), regarding it as a polynomial in $z$ with coefficients dependent on $s$, viz., $D(s, z) = \sum_{k=0}^n d_k(s) z^k$, $n := n_2$, and using the following recursions.

$$zC_{n-1}(s,z) = d_n^\sharp(s) D(s,z) - d_0(s) D^\sharp(s,z), q_{n-1}(s) = 1 \quad (4a)$$

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

$$zC_{m-1}(s,z) = \frac{c_{[m]m}(s) C_m(s,z) - c_{[m]0}(s) C_m^\sharp(s,z)}{q_m(s)}$$

$$q_{m-1}(s) = c_{[m]m}(s) \quad (4b)$$

The division by the $q_m(s)$ is exact. Namely, $q_m(s)$ is a factor of the numerator 2-D polynomial that it divides. The polynomials $C_m(s, z)$, $m = n_2 - 1, \ldots, 0$, are of degrees $(2(n_2 - m)n_1, m)$.

**Theorem 1. [Stability conditions for Algorithm 1.]** *$D(z_1, z_2)$ is stable if, and only if, the following three conditions hold.*

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

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

**(iii)** $\quad \epsilon(s) := C_0(s, z) = c_{[0]0}(s) \neq 0 \quad \forall s \in T \quad (5)$

*where $\epsilon(s)$ is a symmetric polynomial in $s$ (only) and its degree is $2n_1n_2$.*

It can be shown that condition (iii) is replaceable by positivity condition on $T$ of $\epsilon(\tilde{s}) := s^{-M}\epsilon(s)$, $M = n_1n_2$, viz.

$$\epsilon(\tilde{s}) > 0 \quad \forall s \in T \quad . \quad (5')$$

Note that $\epsilon(\tilde{s})$ is real for $s \in T$ because $\epsilon = J\epsilon$.

### 2.3. Companion 1-D stability test

The next algorithm is the basic form for the C-type tests in the classification of the SCMJ tests in [6].

**Algorithm 2.** Assume $p(z)$ (3) assign to it a sequence of polynomials $\{c_m(z),\quad m = n-1, \ldots, 0\}$, where $c_m(z) = \sum_{i=0}^m c_{m,i} z^i$ as follows.

$$zc_{n-1}(z) = p_n^\star p(z) - p_0 p^\sharp(z)\,;\ q_{n-1} = 1 \quad (6a)$$

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

$$zc_{m-1}(z) = \frac{c_{m,m} c_m(z) - c_{m,0} c_m^\sharp(z)}{q_m}\,;\ q_{m-1} = c_{m,m} \quad (6b)$$

**Remark 1.** Algorithm 1 corresponds to applying Algorithm 2 to $p_{\tilde{s}}(z) = D(\tilde{s}, z) := s^{-n_1/2} D(s, z)$ regarded as a polynomial in $z$ with coefficients dependent on $s$ (forming, so called, 'balanced polynomials' - polynomials that extend to equal degree in $s$ and $s^{-1}$ or $s^{1/2}$ and $s^{-1/2}$) and assumed to take values $s \in T$. This way, conjugation of a scalar $c_{m,k}$ corresponds to reversion of the vector $c_{[m]k}$. Nevertheless the substitution of $\tilde{s}$ by $s$ as done in Algorithm 1 is permitted. It can be shown that both choices yield the same sequence of matrices $\{C_m, m = n-1, \ldots, 0\}$ (cf. Theorem 3 below). As a matter of fact, a matricial form of the algorithm dropping all variables (following the approach used for the 2-D tabular test in [3]) provides the most transparent presentation for programming this tabular test. However the polynomial interpretation is the more instrumental presentation for deducing Theorem 1 from the next Theorem 2 and subsequently for simplifying the test by interpolation.

**Theorem 2. [Zero location for Algorithm 2.]** *If Algorithm 2 does not terminate prematurely then $p(z)$ has (no zeros on $T$) $\nu$ zeros in $V$ and $n-\nu$ zeros in $U$, where $\nu$ is given by number of sign variations in the leading coefficients sequence*

$$\nu = Var\{1, c_{n-1,n-1}, c_{n-2,n-2}\ ,\ \ldots\ ,\ c_{0,0}\} \quad (7)$$

A proof for Theorem 2 is available in [6].

**Remark 2.** It is also proved in [6] that the leading coefficients of the sequence of polynomials in Algorithm 1, $\{c_{m,m}\ ,\quad m = n-1, \ldots, 0\}$ constitute the principal minors of the Schur-Cohn matrix of $p(z)$. Evidently, Algorithm 2 does not terminate prematurely if and only if all the principal minors are not vanishing (the Schur-Cohn matrix is strongly regular).

The relation of the modified Jury's test with the principal minors of the Schur-Cohn Bezoutian were used, in combination with Siljak's simplification of testing positivity of the polynomial Bezoutian [7], to prove the single positivity condition in [5]. A proof for Theorem 1 can similarly be obtained, after adding to this batch the simplification for the condition (2) obtained by Huang and Stintzis. (The Huang-Strintzis simplification is a starting point for virtually all 2-D stability tests, see [1, Theorem 5(2,3)], [2] or [3]). Siljak's simplification admits skipping the examinations of the first $n-1$ conditions in the next collection of necessary conditions for 2-D stability.

$$c_{[m]m}(s) \neq 0 \quad \forall s \in T \quad , \quad m = n-1, \ldots, 0 \qquad (8)$$

or the equivalent 'positivity' conditions

$$c_{[m]m}(\tilde{s}) > 0 \quad \forall s \in T \quad , \quad m = n-1, \ldots, 0 \qquad (8')$$

The fact that these are necessary conditions for stability follows from the correspondence between the set $c_{m,m}, m < n$ in Algorithm 2 and the symmetric vectors $c_{[m]m}$ in Algorithm 1. $c_{[m]m}(\tilde{s}) = s^{-(n_2-m)n_1} c_{[m]m}(s)$ are real for $s \in T$ because $c_{[m]m}$ are symmetric vectors. In fact, $c_{[m]m}(\tilde{s})$, $m = n-1, \ldots, 0$, are the principal minors of the Schur-Cohn Bezoutian for $p_{\tilde{s}}(z) := s^{-n_1/2} D(s,z)$ and in particular $\epsilon(\tilde{s})$ is its determinant.

It is interesting to estimate the cost of computation for this 2-D stability test. An approximate count of operations is carried out next that keeps only leading terms in a polynomial expression for the precise count (all counts are of polynomial order). The notation $O(n_{1,2}^k)$ is used here and throughout to indicate that terms with powers $n_1^{\alpha_1} n_2^{\alpha_2}$, $\alpha_1 + \alpha_2 \leq k$ are discarded. Recall that convolution/multiplication of two vectors/polynomials of length/degree $\ell_1$ by $\ell_2$ requires approximately $\ell_1 \times \ell_2$ operations. Similar count of operations is required approximately also for deconvoloution/division of a polynomial of degree $\ell_1 + \ell_2$ by its factor of degree $\ell_1$ or $\ell_2$. Step $k$ of Algorithm 1 requires for each of the $(n_2 - k + 1)$ columns of the matrix $C_{n_2-k}$ two convolutions of $2kn_1$ by $2kn_1$ followed by deconvolution of $4kn_1$ by $2kn_1$. The result is approximately $\sum_{k=1}^{n_2} (2 \times 2 + 4 \times 2) k^2 n_1^2 (n_2 - k)$ operations that after neglecting lower power terms gives $\frac{4}{3} n_1^2 n_2^4 + O(n_{1,2}^5)$. The arithmetic costs associated with the examination of conditions (i) (ii) (iii) in Theorem 1 are by comparison of negligible orders of $O(n_1^2)$, $O(n_2^2)$ and $O(n_1^2 n_2^2)$, respectively. Thus the total cost of the 2-D tabular test is approximately $\frac{4}{3} n_1^2 n_2^4 + O(n_{1,2}^5)$.

The paper [5] emphasizes the fact that the test ends with a single 'positivity test' for a symmetric polynomial of degree $2n_1n_2$ compared to a much higher degree polynomial in previously proposed tabular tests. Indeed, the 2-D test of Maria and Fahmy [8], that was the first proposed tabular test, as well as subsequent tabular tests, including several previous works by Hu and his coauthors (listed in [5]), all end with a (symmetric) polynomial of degree $2n_2 2^{n_1}$. (Most works mention half of this degree because a symmetric polynomial can always be 'folded' into a not symmetric polynomial of half degree. In our view, this extra operation is counterproductive - it increases arithmetic cost, degrades numerical accuracy without achieving any apparent added value.) According to the above cost analysis, the more significant achievement of this tabular test is in reducing the complexity of tabular 2-D stability tests from previous severe exponential complexities (rightly criticized in [1] as impractical for all but testing the simplest filters) to a $O(n^6)$ complexity (say $n_1 = n_2 = n$).

## 3. THE INTERPOLATION PROBLEM

One notable fact that emerges from the above cost analysis is that the complexity of the tabular test is dominated by the $O(n^6)$ cost of the table's construction. The $O(n^4)$ cost of testing the single positivity condition (5) (and even the cost of testing *all* the positivity tests in (8), that requires $O(n^5)$ operations) is negligible by comparison to the demands of Algorithm 1. Another notable observation that follows from Theorem 1 is that the only role that the construction of the table serves is to obtain its last entry, the polynomial distinguished by the notation $\epsilon(s)$.

We show next that it is possible to obtain $\epsilon(s)$ and maintain an overall $O(n^4)$ complexity by *tele*scoping (bringing forth) the last entry of the 2-D table by inter*polation* without its full construction. Telepolation was already presented in [10] to simplify our immittance tabular 2-D stability tests [9]. (The name *immittance* has been given to algorithms that stem from the zero location test formulation in [11] and similar to this test's advantage over the SCMJ class of stability tests, yield algorithms of improved efficiency for several related *scattering* classical algorithms. The immittance formulation replaces two-term recursions in the classical algorithms by three-term recursions and exploit intrinsic symmetry in the problems).

Denote the entries of the coefficient vector of $\epsilon(s)$ by $\epsilon = [\epsilon_0, \ldots, \epsilon_{2M}]^t$. Let

$$\theta = \frac{2\pi}{2M+1} \quad , \quad w = e^{j\theta}$$

where $j = \sqrt{-1}$. If $\epsilon(\tilde{s})$ is known at $M+1$ values $s_i \in T$ given by

$$b_i = \epsilon(\tilde{s}_i) \quad , \quad s_i = w^{-M+i} \quad , \quad i = 0, 1, \ldots, M$$

Then it can be determined from these values by the next expression.

$$\epsilon_{M-m} = \frac{b_M + 2\sum_{k=1}^{M} b_{M-k} \cos(mk\theta)}{(2M+1)} \, , \, m = 0, \ldots, M$$

$$\epsilon_{M+m} = \epsilon_{M-m} \qquad , \quad m = 1, \ldots, M \qquad (9)$$

The derivation of this expression, that follows from a DFT-like formula specialized to the current problem, will become available in a forthcoming journal paper. The cost of determining $\epsilon$ from $n_1n_2 + 1$ values of $b_i = \epsilon(\tilde{s}_i)$ is $n_1^2 n_2^2$ real operations. The division by $2M+1$, that provides an exact reproduction of $\epsilon(s)$, is not necessary for examining the condition (5).

The required values $b_i = \epsilon(\tilde{s}_i)$ are obtained by application of Algorithm 2 to $p_{s_i}(z) = D(s_i, z)$ in accordance with Remark 1 and the following further result.

**Theorem 3.** *Algorithm 1 and its accompanying stability conditions in Theorem 1 may be regarded as projection on $T$ of the 2-D tabular stability test (Algorithm 2 and Theorem 2) in the sense explained in Remark 1. Specifically:*
*(a:) Assume Algorithm 2 is applied to the 1-D polynomial $p_{s_i}(z) = D(s_i, z)$, $s_i \in T$, then $c_{0,0} = \epsilon(\tilde{s}_i) = s_i^{-M} c_{[0]0}(s_i)$.*
*(b:) Stability of the 1-D polynomial $p_{s_i}(z)$ is necessary condition for stability of $D(z_1, z_2)$.*

Proof: The action of Algorithm 1 on $D(\tilde{s}, z)$ for each point $s \in T$ is the same as the action of Algorithm 2 over

$p_{\tilde{s}}(z) = D(\tilde{s}, z) := s^{-n_1/2} D(s, z)$ as was said in Remark 1. Statement (a) is true because it is easily checked that applying Algorithm 2 either to $p_s(z) = D(s, z)$ or to $p_{\tilde{s}}(z) = s^{-n_1/2} D(s, z)$ yields the same $c_m(z)$ for $m = n - 1$ and thus both initiations create the same sequences $\{c_m(z), m = n-1, \ldots, 0\}$. Therefore $c_{0,0} = \epsilon(\tilde{s}_i)$. Statement (b) follows from (2).

## 4. THE PROPOSED 2-D STABILITY TEST

A possible implementation of the emerging new test is summarized as a 4 steps procedure below. 'Exit' is used to mark points that admit early termination of the procedure because an indication that "$D(z_1, z_2)$ is not stable" has already been found.

### A 2-D Stability Test Procedure for $D(z_1, z_2)$

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

**Step 2.** Set $M = n_1 n_2$, $\theta = \frac{2\pi}{2M+1}$, $w = e^{j\theta}$.

For $i = 0, 1, \ldots, M$ do: Set $s_i = w^{-M+i}$. Apply the companion 1-D stability (Algorithm 2 + Theorem 2) to $p_{s_i}(z) = D(s_i, z)$. As soon as a $c_{m,m} \leq 0$ $m = 1, 2, \ldots$ is detected ($p_{s_m}(z)$ is not 1-D stable) - 'exit'. Otherwise, retain the last element as $b_m := c_{0,0} (> 0)$.

**Step 3.** Use (9) to obtain $\epsilon(s) = \sum_{i=0}^{2M+1} \epsilon_i s^i$ from the values $b_i$ $i = 0, \ldots, M$.

**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'.

The paper is not self contained only in not showing how to carry out step 4. For the integrity of the current approach one might want to use the companion test (Algorithm 2 and theorem 2) for this task as well. This is possible in principle but requires extension of the algorithm to a full zero location method (one that works also in the not strongly regular case) that at this time has not yet been published. Anyway, from the point of view of minimal operations, the most economical tests in the SCMJ class are not C-type tests but B-type tests [6]. Raible's test [12] is a B-type test and it also provides means to overcome singular cases. Raible's test performs step 4 in $2n_1^2 n_2^2 + O(n_{1,2}^3)$ real operations.

## 5. COST EVALUATION

Let us carry out an approximate count of operations for the above procedure. The count is in terms of real arithmetics. Step 1 is a 1-D stability test for a polynomial of degree $n = n_1$. Its $O(n^2)$ complexity is negligible. Step 2 involves $n_1 n_2$ times the application of Algorithm 2 to a complex 1-D polynomials (plus the real polynomial $P_{s_0}(s)$) each of degree $n_2$. It can be carried out in $3n_1 n_2^3 + O(n_{1,2}^3)$ real operation. Step 3 requires $n_1^2 n_2^2$ real operations. The testing of condition in step 4 by Raible's test requires $2n_1^2 n_2^2 + O(n_{1,2}^3)$ real operations. The summation of the above counts yields an overall complexity for the test of $3n_1^2 n_2^2 + 3n_1 n_2^3 + O(n_{1,2}^3)$ real operations.

## 6. CONCLUSIONS

A tabular 2-D stability test for discrete-time systems by Hu and Jury was revisited and simplified via telepolation. Telepolation reduces the complexity of the test from $O(n^6)$ to a very low $O(n^4)$ of operations. With telepolation, the 2-D stability of a polynomial of degree $(n_1, n_2)$ is carried out by a set of $n_1 n_2 + 1$ 1-D stability tests of degree $n_1$ or $n_2$ plus one zero location test of a 1-D polynomial of degree $2n_1 n_2$. Another benefit drawn from this structure stems from the fact that all necessary conditions for 1-D stability are also necessary condition for 2-D stability. The ample of early indications for an unstable polynomial lowers even further the cost wasted on an unstable polynomial.

## 7. REFERENCES


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

[2] 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.

[3] Y. Bistritz, "Stability Testing of Two-Dimensional Discrete Linear System Polynomials by a Two-Dimensional Tabular Form" *IEEE Trans. on Circuits and Systems, part I* , Vol. 46, pp. 666-676, June 1999.

[4] E. I. Jury, "Modified Stability Table for 2-D Digital Filter", *IEEE Trans. on Circuits and Systems*, vol. CAS-35, 116-119, 1988.

[5] X. Hu and E. I. Jury, "On Two-Dimensional Filter Stability Test" *IEEE Trans. on Circ. Syst.* vol. CAS-41, pp. 457-462, 1994.

[6] Y. Bistritz, "Reflection on Schur-Cohn Matrices and Jury-Marden Tables and Classification of Related Unit Circle Zero Location Criteria" *Circuits Systems Signal Process*, Vol. 15, pp. 111-136, 1996.

[7] D. D. Siljak, "Stability criteria for two-variable polynomials", *IEEE Trans. on Circuits and Systems*, Vol. CAS-22, pp. 185-189, 1975.

[8] 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, 1973.

[9] Y. Bistritz, "Stability Test for 2-D LSI Systems Via a Unit Circle Test for Complex Polynomials" in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 789-792, Seattle, Washington, 1995.

[10] Y. Bistritz, "A Stability Test of Reduced Complexity for 2-D Digital System Polynomials" *Proc. IEEE Int. Symp. Circuits Syst.*, Monterey, CA, May 1998.

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

[12] R. H. Raible, "A simplification of Jury's tabular form" *IEEE Trans. on Automat. Control*, vol. AC-19 pp. 248-250, 1974.