IEEE Int. Symp. Circuits Syst., ISCAS-95 vol. 2, pp. 789-792, Seattle, Washington, May 1995.

Stability Test for 2-D LSI Systems Via a Unit Circle Test for Complex Polynomials

Yuval Bistritz

Abstract— **A new algebraic test for two-dimensional digital filters is developed based on the author's stability test for one-dimensional discrete system polynomials with complex coefficients. The new method consists of an array of polynomials and an accompanying set of necessary and sufficient conditions for stability. Programming the construction of the array is simple and the execution involves a lower count of computation then reported for previous tests. Testing the stability conditions needs just a single "positivity test" of the last polynomial in the array plus standard 1-D stability conditions. A larger set of conditions necessary for stability that may be useful in other modes of application is also provided.**

I. Introduction

A two-dimensional (2-D, bivariate) polynomial

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

is said to be stable if

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

where

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

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

The problem under consideration is to determine whether a given $D(z_1, z_2)$ is stable. This problem is the key for the stability determination of 2-D linear shift-invariant systems and filters.

This paper develops a 2-D stability test that is based on the 1-D unit circle test for polynomials with complex coefficients in [3]. The proposed test is *algebraic* namely, it determines whether $D(z_1, z_2)$ is stable or not in a finite number of arithmetic operation. The test consists of a sequence of matrices - the "2-D table" - that is obtained by a three-term recursion of 2-D polynomials, and a set of stability conditions posed on it that has to be examined. This set of stability conditions is reduced to one 1-D test and one test for positivity of a polynomial on T. A simplification of this type was introduced by Siljak who showed in [8] that for determining positive definiteness of a the Schur-Cohn polynomial matrix over the unit circle it suffices to determine definitness at a point and the positivity of the determinant polynomial on the unit circle. In a tabular test it was adopted to 2-D stability using the Jury-Marden table [6] The initial form of the 2-D table is simplified by removing from it common factors that are shown to be redundant. Combination of the simplified table form and stability conditions achieves an overall cost of computation reduced from an initial exponential order typical to earlier 2-D table tests [7] to an order n^6 (say $n = n_1 = n_2$).

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

II. Background Theory and Notation

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

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

Here p_k is the $k+1$-th column of P, and $p_k(s) = \mathbf{s}^t p_k$, $k = 0, \ldots, n$, are the coefficient polynomials of $P(s, z)$ regarded as a 1-D polynomial in the second variable z.

A standard approach to testing 2-D stability is to regard the 2-D polynomial $D(z_1, z_2)$ as a polynomial in z_2 with coefficients that are polynomials in the variable z_1 and then use an 1-D stability test in conjunction with one of several stability condition that relax the condition (2) by a search of (z_1, z_2) over $T \times \bar{V}$ (plus a 1-D tests).

Lemma 1. *$D(z_1, z_2)$ is stable if and only if*
(i) $D(z, a) \neq 0$ for all $z_1 \in \bar{V}$ and some $a \in \bar{V}$
(ii) $D(s, z) \neq 0$ for all $(s, z) \in T \times \bar{V}$.

This result was obtained by Strintzis [9]. It is often called Huang's theorem who obtained it first in [5] for $a = \infty$. A list of several such simplifying conditions may be found in [6] and in [4] where the latter also contains alternative proofs and further extensions. Note that part (a) is a simple 1-D polynomial stability test. Thus the task of an algebraic 2-D stability test concerns essentially an efficient way for testing the condition (ii).

Given a polynomial $P(s, z)$ it will be actually more useful to think of the coefficient matrix P as related to a different function as follows.

$$P(\tilde{s}, z) = s^{-m/2} P(s, z) = \sum_{k=0}^{n} p_k(\tilde{s}) z^k = \tilde{\mathbf{s}}^t P \mathbf{z}$$

where $\tilde{\mathbf{s}} := [s^{-m/2}, s^{-(m-1)/2}, \ldots, s^{(m-1)/2}, s^{m/2}]^t$, and $\tilde{s}$ as a function argument represent power series in (s^{-1}, s) (in the context of this paper m will be an even integer) of equal length as shown. We shall refer to such polynomials as "*balanced polynomials*".

Superscript $\sharp$ will denote (conjugate) reversion, defined for a matrix and a vector by

$$P^\sharp = JP^\star J \qquad p^\sharp = Jp^\star \quad ,$$

where J denotes the reversion matrix with 1's on the main anti-diagonal and zeros and $\star$ denotes complex conjugation.

Polynomial multiplication corresponds to convolution of their coefficient vectors. Convolution will be denoted by $*$ thus

$$h = f * p_i \longleftrightarrow h(s) = f(s)p_i(s)$$

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

$$G = f * P = [f * p_0, f * p_1, \ldots, f * p_n] = [g_0, g_1 \cdots g_n] \longleftrightarrow G(s,z) = f(s)P(s,z)$$

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

$$P = f \setminus G = [p_0, p_1 \ldots, p_n] \longleftrightarrow$$

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

and it will represent extraction of a factor common to all the polynomial coefficients.

We shall form for the tested polynomial a sequences of bivariate polynomials of the form $\{F_m(\tilde{s}, z)\}$, where $F_m(s,z)$ will be of degree $\ell_f(m)$ in s and of degree $n-m$ in z. Equivalently it may be said that we form for the tested polynomial a sequence of real matrices $\{F_m\,,\,0,\ldots,n\}$ (that will also be called the 2-D 'table'). The matrices F_m will all be centro-symmetric, namely $F_m^\sharp = F_m$. The sequential index m of the matrix F_m, when referring to its columns (and rows), will be set in brackets and precede the other indices. For example we may write

$$F_m(\tilde{s}, z) = [f_{[m]\ 0}(\tilde{s})\ ,\ f_{[m]\ 1}(\tilde{s})\ ,\ldots\ ,\ \ldots, f_{[m]\ n-m}(\tilde{s})]\mathbf{z} =$$

$$\tilde{\mathbf{s}}^t[f_{[m]\ 0}\ ,\ f_{[m]\ 1}\ ,\ \ldots, f_{[m]\ n-m}]\mathbf{z}$$

where $f_{[m]\ k}$ is the $k+1$-th column of F_m,

$$f_{[m]\ k} = [f_{[m]0,k}\ ,\ f_{[m]1,k}\ ,\ \ldots, f_{[m]\ell_f(m),k}]^t \quad .$$

III. Division-free 1-D Stability Test

A univariate polynomial $P_n(z)$ is said to be stable if it has all its zeros in U. This stability condition may be written in a form closer to (2),

$$P_n(z) \neq 0 \quad \text{for} \quad z \in \bar{V} \tag{3}$$

We present next a modification for the stability testing procedure in [3] that avoids the arithmetic operation of division. Assume

$$P_n(z) = \sum_{i=0}^{n} p_i z^i \quad , \quad P_n(1) \neq 0 \text{ is real.} \tag{4}$$

Algorithm 1 : Division-Free 1-D Complex Array Consider the polynomial $P_n(z)$ (4) and form for it a sequence $\{F_m(z)\,,\ m = 0, 1, \ldots, n\}$ as follows:

(i) **Initiation.**

$$F_0(z) = P_n(z) + P_n^\sharp(z)\ ,\ F_1(z) = \frac{P_n(z) - P_n^\sharp(z)}{(z-1)}$$

(ii) **Body.** For $m = 0, \ldots, n-2$ obtain $F_{m+2}(z)$:

$$zF_{m+2}(z) = (f_{m,0}f^\star_{m+1,0} + f^\star_{m,0}f_{m+1,0}z)F_{m+1}(z)$$

$$-f_{m+1,0}f^\star_{m+1,0}F_m(z)$$

The algorithm is considered *normal* if all $f_{m,0} \neq 0$. Normal conditions are necessary conditions for stability. When they hold, each $F_m(z)$, $m = 0, 1, \ldots n$, is a conjugate-symmetric polynomial of degree $n - m$ viz.,

$$F_m(z) = \sum_{i=0}^{n-m} f_{m,i} z^i,\ F_m^\sharp(z) = F_m(z) \quad (f_{m,n-m-i} = f^\star_{m,i})$$

It also follows then that all $F_m(1)$ are real.

Theorem 1. (Stability Criterion for Algorithm 1.) *Assume $P_n(z)$ (4) and that Algorithm 2 is applied to it. $P_n(z)$ is stable if and only if*

$$Sgn\{F_m(1)\} = Sgn\{P_n(1)\} \quad m = 0, 1, \ldots, n \quad .$$

Proof. Comparison of the recursions in Algorithm 1 with the recursion for the sequence $\{T_m(z)\}$ in [3] reveals that the relation between the sequences $\{F_m(z)\}$ and $\{T_m(z)\}$ is

$$F_m(z) = \psi_m T_m(z) \text{where} \quad \psi_0 = 1\ ,\ \psi_1 = 1$$

$$\psi_m = |f_{m-1,0}|^2 \psi_{m-2} = |t_{m-1,0}|^2 \psi^2_{m-1}\psi_{m-2} \quad ,\quad m \geq 2$$

The proof now follows from the stability theorem in [3] because all the ψ_m are positive. □

Suppose we tried to test condition (ii) in Lemma 1 by applying to it the 1-D test in [3] viewing $D(s,z)$ as a polynomial in z that has coefficients that are polynomials in s (assume for a moment for the sake of this argument that $D(s,1)$ is real for all $s \in T$ as needed in this 1-D test). It becomes apparent that this approach would involve a manipulation of a sequence of polynomials in z with rational-function coefficient. In difference a similar extension of the above modified 1-D test, will circumvent the treatment of rational function coefficients. A subsequent need to testing positivity conditions (over $s \in T$) will also be simplified as it will be posed on polynomials rather than on rational functions.

IV. A Basic 2-D Test

The polynomial $D(s,z)$ is not directly suitable for Algorithm 1 because $D(s,z)$ at $z=1$ is not real for all $s \in T$. Following the suggestion in [3] we therefore form first an auxiliary bivariate function

$$M(s^{-1},s,z) := D(s^{-1},1)D(s,z) = M(\tilde{s},z) = \tilde{\mathbf{s}}^t M\mathbf{z} \quad (5)$$

Now, $M(\tilde{s},1)$ is real and positive for $s \in T$ and we may check whether condition (ii) of Lemma 1 holds by the following equivalent condition

$$M(\tilde{s},z) \neq 0 \text{ for } s \in T, z \in \bar{V}. \quad (6)$$

A. Array Construction for M(s̃,z)

The initial 2-D stability table is obtained by applying Algorithm 2 to $M(\tilde{s},z)$ regarding it as a polynomial of degree $n = m_2$ in z with balanced polynomial coefficients dependent on $\tilde{s}$

Algorithm 2: Array for M(s̃,z). Form for $M(\tilde{s},z)$ a sequence $\{F_m(\tilde{s},z) = \sum_0^m f_k(\tilde{s})z^k\ ,\ m = 0,1,\ldots,n\}$, as follows.

(i) **Initiation.**

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

(ii) **Body.** For $m = 0,1,\ldots,n-2$ obtain $F_{m+2}(\tilde{s},z)$ by:

$$h_m(\tilde{s}) = f_{[m]\ 0}(\tilde{s}) f^\sharp_{[m+1]\ 0}(\tilde{s})$$

$$r_{m+1}(\tilde{s}) = f_{[m+1]\ 0}(\tilde{s}) f_{[m+1]\ 0}(\tilde{s})^\sharp$$

$$zF_{m+2}(\tilde{s},z) =$$

$$= (h_m(\tilde{s}) + h^\sharp_m(\tilde{s})\ z\ )F_{m+1}(\tilde{s},z) - r_{m+1}(\tilde{s})F_m(\tilde{s},z)$$

Note that for $s \in T$ complex conjugation correspond to reversion of the coefficient polynomials, e.g.,

$$f_{[m]\ k}(\tilde{s}) = f_{[m]\ k}(s^{-1},s) \longleftrightarrow f_{[m]\ k} \quad \Longleftrightarrow$$

$$f_{[m]\ k}(s^{-1},s)^* = f_{[m]\ k}(s,s^{-1}) = \tilde{\mathbf{s}}Jf_{[m]\ k} \longleftrightarrow f^\sharp_{[m]\ k}$$

The (conjugate-) symmetry of the 1-D polynomials in Algorithm 1 transforms currently into centro-symmetry, $F_m = JF_mJ = F^\sharp_m$, of the matrix of coefficients in the sequence produced by Algorithm 2. This symmetry can be exploited to actually compute only half of the entries of each F_m.

B. Stability Criteria

2-D Stability conditions for Algorithm 3 can be deduced at once from Theorem 1 and Lemma 1.

Theorem 2. (Stability conditions for Algorithm 2.) *Necessary and sufficient conditions for $D(z_1,z_2)$ to be stable are*

(i) $D(z,1) \neq 0$ for $z \in \bar{V}$
(ii) $D(1,z) \neq 0$ for $z \in \bar{V}$
(iii) $\varphi_m(\tilde{s}) \neq 0$ for $s \in T\ , m = 0,1,\ldots,n$

where $\{F_m(\tilde{s},z)\}_0^n$ is the array produced by Algorithm 2 for $M(\tilde{s},z)$ and $\varphi_m(\tilde{s}) := F_m(\tilde{s},1)$.

C. Refined 2-D Stability Conditions

A less expected result than the Theorem 2 is that it suffices to examine a single "positivity test" in using Algorithm 2. An appropriate proof for the next assertion will become available in a forthcoming publication.

Theorem 3. (Sharper stability conditions for Algorithm 2.) *Necessary and sufficient condition for $D(z_1,z_2)$ to be stable are*

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

where $\varphi_n(\tilde{s}) = F_n(\tilde{s},1) = f_{[n]\ 0}(\tilde{s})$.

V. A Reduced Testing Procedure

As a matter of fact Algorithm 2 generates 2-D polynomials with multiple $\tilde{s}$ polynomial factors. We first characterize this feature and then devise a refined algorithm that is freed from these factors. Afterwards necessary and sufficient conditions for stability using the reduced array obtained and show that the eliminated factors are indeed redundant. The removal of these redundant factors contributes most significantly to the efficiency of the final algorithm. It can be shown to reduce overall cost from an arithmetic count that increases exponentially with the degree n (say $n_1 = n_2 = n$) of the tested polynomial to a count of order n^6. The results of this section (as well as the previous theorem) are stated without proof due to space limitations. Adequate proofs will be made available in forthcoming publications.

A. Redundant Factors

The next Lemma spots common $\tilde{s}$ factors in the sequence $\{F_m(\tilde{s},z)\}$ and characterizes the way they build up.

Lemma 2. *Consider the sequence $\{F_m(\tilde{s},z)\}_0^n$ produced by the Algorithm 2.*
(a) If $f(\tilde{s})$ is a factor of $F_m(\tilde{s},z)$ $m \geq 0$ then it is a factor of all subsequent $F_{m+i}(\tilde{s},z)$ $i \geq 1$.
(b) For any adjacent two polynomials $G_0(\tilde{s},z)$, $G_1(\tilde{s},z)$ of degrees k and $k-1$ in z, respectively ($3 \leq k \leq n$), let $G_2(\tilde{s},z)$, $G_3(\tilde{s},z)$ be the two consecutive polynomials generated by Algorithm 2. Then $g_{[1]\ 0}(\tilde{s})g_{[1]\ 0}(\tilde{s}^{-1})$ is a factor of $G_3(\tilde{s},z)$. Namely, this balanced symmetric polynomial divides with no reminder $g_{[3]\ i}(\tilde{s})$ for all $i = 0,1,\ldots,k-3$.

Lemma 2 and the structure of the recursion in Algorithm 2 admit a change in the algorithm such that a sequence, $\{E_m(\tilde{s},z)\}$, that is clean from these factors will be obtained directly by a modified recursion.

Algorithm 3: Reduced Array for M(s̃,z) Construct for $M(\tilde{s},z)$ a sequence of polynomials $\{E_m(\tilde{s},z) = \sum_0^m e_k(\tilde{s})z^k\ ,\ m = 0,\ldots,n\}$ defined by
Initiation.

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

Body. For $m = 0, 1, \ldots, n-2$ obtain $E_{m+2}(\tilde{s}, z)$ by:

$$g_m(\tilde{s}) = e_{[m]\ 0}(\tilde{s})e^{\sharp}_{[m+1]\ 0}(\tilde{s})$$

$$q_{m+1}(\tilde{s}) = e_{[m+1]\ 0}(\tilde{s})e^{\sharp}_{[m+1]\ 0}(\tilde{s})$$

$$zE_{m+2}(\tilde{s}, z) =$$

$$= \frac{(g_m(\tilde{s}) + g^{\sharp}_m(\tilde{s})z)E_{m+1}(\tilde{s}, z) - q_{m+1}(\tilde{s})E_m(\tilde{s}, z)}{\rho_m(\tilde{s})}$$

where $\rho_0(\tilde{s}) = 1$, $\rho_m(\tilde{s}) = q_m(\tilde{s})$, $m \geq 1$

B. Stability Criteria for the Reduced Table

The construction of the array $\{E_m(\tilde{s}, z)\}$ is significantly more efficient than $\{F_m(\tilde{s}, z)\}$ because, as it may be shown, in the new array the row sizes increase only linearly with m. The question is whether the use of this reduced array does not increase the number or complicate the cost of the stability conditions that need to be posed on them. The next theorems state that stability conditions in form identical to those accompanying Algorithm 3 remain valid also for $\{E_m(\tilde{s}, z)\}$. Of course, an identical number and form of necessary and sufficient conditions, when posed on $\{E_m(\tilde{s}, z)\}$, are examined at a lower cost of computation, because the overall reduction in row sizes also means that they are posed on lower degree polynomials.

Theorem 4. (Stability conditions for Algorithm 3.) *Necessary and sufficient conditions for $D(z_1, z_2)$ to be stable are*

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

where $\{E_m(\tilde{s}, z)\}_0^n$ is the array produced by Algorithm 3 for $M(\tilde{s}, z)$ and $\epsilon_m(\tilde{s}) := E_m(\tilde{s}, 1)$.

This is the parallel of Theorem 2. Sharper conditions that parallel Theorem 3 also hold:

Theorem 5. (Sharper stability conditions for Algorithm 3.)
Necessary and sufficient condition for $D(z_1, z_2)$ to be stable are

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

where $\epsilon_n(\tilde{s}) = E_n(\tilde{s}, 1) = e_{[n]\ 0}(\tilde{s})$.

VI. Testing Procedure Summary

So far we used polynomial notation because it provides a compact way to write the recursions etc., and we also find it a constructive tool for the derivation of the procedure. At the same time, the notation we introduced provides a quick translation to alternative semi or full matricial notation. The matricial interpretation becomes the more useful choice for the programming of the new test. The implementation becomes in particular straightforward by an array-oriented language like MATLAB. For this end note for example that the main recursion step may also be written in the next way,

$$[0, E_{m+2}, 0] = \rho_m \setminus (\hat{E}_{m+1} + \hat{E}^{\sharp}_{m+1} - q_{m+1} * E_m)$$

where $\hat{E}_m := g_{m-1} * [E_m, 0]$. Similarly, in order to obtain M from D as in (5), let the coefficient vector of $D(s, 1)$ be $b = \sum_{i=0}^{n} d_i$. Then, $M = b^{\sharp} * D$. (Convolution and deconvolution are standard routines in MATLAB.) An outline for testing whether $D(z_1, z_2)$ is stable may be summarized as the following:

step 1: Test whether $D(z_1, 1)$ is stable. If stable go to Step 2 else $\Longrightarrow D(z_1, z_2)$ is not stable ('exit').

step 2: Test stability of $D(1, z_2)$. If stable go to Step 3 else $\Longrightarrow D(z_1, z_2)$ is not stable ('exit').

step 3: Form $M(\tilde{s}, z)$ and apply to it Algorithm 3. [Optionally, test whether $E_m(\tilde{s}, 1) \neq 0$ right after each new $E_m(\tilde{s}, z)$ is formed. If not $\Longrightarrow D(z_1, z_2)$ is not stable ('exit').] Test whether $\epsilon_n(\tilde{s}) \neq 0$. If true (after passing steps 1 & 2) $\Longrightarrow D(z_1, z_2)$ is stable, else it is not stable.

Testing 1-D stability may be done by Algorithm 1, but is more efficiently done by the test in [2]. The latter reference also provides the efficient means (see 'Type I singularity' in there) to testing algebraically the condition $\epsilon(\tilde{s}) \neq 0$ on T (valid iff $\epsilon(s)$ has half of its zeros in V and their reciprocal in U).

The option stated in step 4 suggests to add to the recursion the checking the positivity of each $\varphi_m(\tilde{s})$ that are further necessary conditions for stability. This approach may save computation in a long run for some application by allowing exit from the construction of the rest of the table with an early indication of instability.

References

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

[2] 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.

[3] Y. Bistritz, "A circular stability test for general polynomials" *Systems & Control Letters*, vol. 7, pp. 89-97, April 1986.

[4] P. Delsarte, Y. Genin and Y. Camp "A simple proof of Rudin's multivariable stability theorem" *IEEE Trans. Acoust. Speech and Sig. Proc.* Vol. ASSP-28, pp.701-704, Dec. 1890.

[5] T. S. Huang, "Stability of two-dimensional recursive filters", *IEEE Trans. Audio Electroacous.*, Vol. AU-20, pp. 158-163, June 1972.

[6] E. I. Jury, "Stability of Multidimensional Systems and Related Problems" *Multidimensional Systems: Techniques and Applications*, S. G. Tzafestas Ed., New York:Marcel Dekker,1986, Ch. 3.

[7] B. T. O'Connor and T. S. Huang, "Stability of general two-dimensional recursive digital filters", *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-26, pp. 550-560, December 1978.

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

[9] M. G. Strintzis, "Test of stability of multidimensional filters", *IEEE Trans. on Circuits and Systems*, Vol. CAS-24, pp. 432-437, August 1977.