Stability Testing of 2-D Digital System Polynomials Using a Modified Unit Circle Test for 1-D Complex Polynomials

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

Abstract

A new algebraic test for determining whether all zeros of a two-variable ('two- dimensional', 2-D) polynomial reside in the interior of a unit bi-circle is developed. The method provides a stability test for 2-D digital filters and systems. It is based on a modified unit circle zero location test by the author for one variable polynomials with complex coefficients. The test comprises a '2-D table' in the form of a sequence of centro-symmetric matrices and an accompanying set of necessary and sufficient conditions posed on it. The sequence is constructed by a three-term recursion of matrices or two variable polynomials. The set of necessary and sufficient conditions, at its minimal setting, consists of only one "positivity test" plus a standard 1-D stability test. Additional useful stability conditions that 2-D stability implies but that need not be checked to prove 2-D stability are also brought.

1. Introduction

A 2-D (two-dimensional, two-variable) 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

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

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

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(LSI, digital) systems and filters [1]. Introduction and background for 2-D stability is well served by the textbooks [2] [3] and the survey [4].

The paper proposes a new *algebraic* method, based on the 1-D test in [5], to determine, in a finite number of arithmetic operation, whether or not $D(z_1, z_2)$ is stable. The test comprises a sequence of matrices - referred as the "2-D table" - that is obtained by a three-term recursion of 2-D polynomials, and a set of associated algebraic type (finite computation) conditions that have to be examined. The set of stability conditions at its minimal setting involves one 1-D test and one test for positivity of a polynomial on T. The single positivity test is reminiscent of a contribution introduced to this field of 2-D stability by Siljak [6] in a context of Schur-Cohn polynomial matrix. However, in difference from the approach implied in [7, 8], in the current method the single positivity test property emerges from direct and inherent considerations with no reference to the Schur minors or other extraneous results.

The paper is organized as follows. The next section introduces notation and cites the Huang-Strintzis simplification to condition (2) which is the starting point of most reported methods for testing condition (2). In section 3 we modify slightly the stability test in [5] to make it more suitable for the current generalization. Section 4 introduces a first form of the 2-D stability table, the "F-table", and first associates to it an anticipated set of conditions necessary and sufficient for stability. Afterwards, a smaller set of stability conditions with a single positivity test is obtained for this table. Section 5 is devoted to a more efficient 2-D stability test that comprises a reduced size table, the "E-table", and its stability conditions. It begins by revealing and removing redundancy in the size of the F-table, proceeds with obtaining stability conditions for the resulting E-table and ends by comparing E-table test to the F-table test and with a numerical illustration. All theorems are stated without proofs that will become available in a forthcoming full journal version of the paper [9].

2. Background Theory and Notations

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 (the role of the two variables may be interchanged) and then use a 1-D stability test in conjunction with one of several stability conditions that relax the condition (2) by allowing a search of (z_1, z_2) over only $T \times \bar{V}$ (plus a 1-D

1058-6393/96 $5.00 © 1996 IEEE
Proceedings of ASILOMAR-29

tests).

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

$$\text{(a)} \quad D(z,a) \neq 0 \quad \text{forall} \quad z_1 \in \bar{V} \text{ and some } a \in \bar{V} \quad (3)$$

$$\text{(b)} \quad D(s,z) \neq 0 \quad \text{for all} \quad (s,z) \in T \times \bar{V}. \quad (4)$$

This result was introduced to the field of 2-D stability tests by Huang for $a = \infty$ [10] and its above form by Strintzis [11]. Several other such simplifying stability conditions are also known [4, 2, 3]. Note that part (a) of the Lemma 1 is a standard 1-D polynomial stability test. So essentially the task of an algebraic 2-D stability test is to handle efficiently condition (b). Condition (a) may be handled by the well known Schur-Cohn test and Marden-Jury tables. The most efficient 1-D tests available are the tests in [12, 13, 5]. Note that choosing a real value for a offers the advantage of staying within real polynomials for a real D. We shall not repeat this plurality further on and will in all theorems $a = 1$, a value that integrates nicely with the role that $z = 1$ plays in the underlying 1-D stability tests.

For a 2-D polynomial $P(s,z) = \sum_{i=0}^{m}\sum_{k=0}^{n} p_{i,k}s^i z^k$ we shall use $P = (p_{i,k})$ to denote the matrix of its coefficients. 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 respective 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

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

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 polynomial coefficients of $P(s,z)$ regarded as a 1-D polynomial in the second variable z. This notation does not show explicitly the row indices of the entries of $P = (p_{i,k})$. Row indices may be added when needed, $p_k = [p_{0,k}, p_{1,k}, \cdots p_{m,k}]^t$. However, most of the time it will be convenient to discard them because we shall manipulate mostly vectors and act on matrices by columns.

Superscript $\sharp$ will denote (conjugate) reversion, defined for a matrix and a vector, respectively, 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 * denotes complex conjugation.

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

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

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

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

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 polynomials $g_k(s)$.

Given a polynomial $P(s,z)$ it will become 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 denotes power series in the pair of variables (s^{-1}, s) or $(s^{-1/2}, s^{1/2})$ to equal extent as shown. A polynomial in $\tilde{s}$ like $p_k(\tilde{s})$ is called a *"balanced polynomials"*.

The tested polynomial $D(z_1, z_2)$ will be associated with sequences of 2-D polynomials of the form $\{F_m(\tilde{s}, z)\ m = -1, 0, \ldots, n\}$, where $F_m(s,z)$ will of degree $n - m$ in z and of degree some degree $\ell_f(m)$ in s to be determined later. Equivalently, it may be said that D is associated with a sequence of matrices $\{F_m\ ,\ m = -1, 0, \ldots, n\}$ (also to be called the '2-D stability table'). The matrices F_m will all be centro-symmetric, namely $F_m^\sharp = F_m$. When referring to columns (and rows) of F_m, the sequential index m will be distinguished by being 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 $f_{[m]\ k} = [f_{[m]0,k}\ , f_{[m]1,k}\ , \ldots, f_{[m]\ell_f(m),k}]^t$ is the $k+1$-th column of F_m.

3. The Underlying 1-D Stability Test

A 1-D (one-variable) polynomial $P_n(z)$ is said to be stable if it has all its zeros in U. This is the familiar condition for stability of LSI systems and filters where $P_n(z)$ forms the system characteristic polynomial or the Z-transfer function denominator. The stability condition may be rephrased into a form closer to (2) by writing it as

$$P_n(z) \neq 0 \quad \text{for} \quad z \in \bar{V} \quad . \quad (5)$$

The current 2-D stability test is based on the 1-D stability test in [5] which modifies the initial conditions of the test for 1-D complex polynomial in [12, 13] to a form that admits a more efficient generalization to 2-D stability. In principle it is possible to obtain a stability test for $D(z_1, z_2)$ by using the 1-D stability test directly in its form in [5] to test condition (b) in Lemma 1, viewing $D(s,z)$ as a 1-D polynomial in z that has coefficients that are polynomials in s. However this approach would involve the manipulation of a sequence of polynomials in z with rational-function dependent on s coefficients. Rational functions are more difficult to manipulate than polynomials both at first phase of the construction of a generalized stability 'table' as well as at the next phase that raises requirements to test such functions for positivity over T. Inspection on the 1-D test in [5] reveals that such rational function are caused by the division operation involved in the creation of the multipliers δ_m in there. Therefore our first step toward our goal is to devise a division free variant for this test.

Consider a polynomial,

$$P_n(z) = \sum_{i=0}^{n} p_i z^i \quad , \quad \mathcal{R}e\{P_n(1)\} \neq 0 \qquad (6)$$

Algorithm 1 : Division-Free 1-D Table.
Construct for $P_n(z)$ a sequence of polynomials $\{F_m(z), m = -1, 0, 1, \ldots, n\}$ as follows:
(i) **Initiation.** $F_{-1}(z) = (z-1)(P_n(z) - P_n^\sharp(z))$

$$F_0(z) = P_n(z) + P_n^\sharp(z) \qquad (7)$$

(ii) **Body.** For $m = 0, \ldots, n-1$: $zF_{m+1}(z) =$

$$(f_{m-1,0}f^{\star}_{m,0} + f^{\star}_{m-1,0}f_{m,0}z)F_m(z) - f_{m,0}f^{\star}_{m,0}F_{m-1}(z) \quad (8)$$

The algorithm is referred *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., $F_m(z) = \sum_{i=0}^{n-m} f_{m,i} z^i$, $f_{m,n-m-i} = f^{\star}_{m,i}$

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

Theorem 1. (Stability Conditions for Algorithm 1.) *Assume Algorithm 1 is applied to $P_n(z)$ (6). $P_n(z)$ is stable if, and only if,*

$$Sgn\{F_m(1)\} = Sgn\{\mathcal{R}e\{P_n(1)\}\}, \; m = 0, \ldots, n \; . \quad (9)$$

Remark 1. The 'normal conditions' in [5] that are necessary conditions for stability transform here to the condition that all $f_{m,0} \neq 0$. Here too $f_{m,0} = 0$ implies instability but in difference it does not interrupt the recursion. Instability implied by the occurrence of a $f_{m,0} = 0$ is detected by causing (9) not to hold.

4. 2-D Stability Test - First Form

Our goal is essentially to determine an efficient way to test condition (b) of Lemma 1. It is easily realized that condition (b) of Lemma 1 holds if and only if $D(\tilde{s}, z) \neq 0 \, \forall (s, z) \in T \times \bar{V}$. This permits applying the 1-D test to $D(\tilde{s}, z)$ rather then to $D(s, z)$. The gain is that complex conjugation of balanced polynomials for values $s \in T$ retains the length of their coefficient vectors whereas for $D(s, z)$ it would double the row sizes of the two initial matrices.

2-D Table Construction

The initial form of our 2-D stability algorithm is obtained by the application of Algorithm 1 to $D(\tilde{s}, z)$ regarding it as a 1-D polynomial in z with coefficients dependent on $s \in T$.

Algorithm 2: First 2-D table (F-table).
Given the polynomial $D(z_1, z_2)$ to be tested, form a sequence $\{F_m(\tilde{s}, z) = \sum_0^{n-m} f_k(\tilde{s}) z^k \; , m = -1, 0, 1 \ldots, n (= n_2)\}$ by the following recursion.
(i) **Initiation.** $F_{-1}(\tilde{s}, z) = (z-1)(D(\tilde{s}, z) - D^\sharp(\tilde{s}, z))$,

$$F_0(\tilde{s}, z) = D(\tilde{s}, z) + D^\sharp(\tilde{s}, z) \qquad (10)$$

(ii) **Body.** For $m = 0, 1, \ldots, n-1$:

$$h_m(\tilde{s}) = f_{[m-1]\;0}(\tilde{s}) f^\sharp_{[m]\;0}(\tilde{s})$$
$$r_m(\tilde{s}) = f_{[m]\;0}(\tilde{s}) f^\sharp_{[m]\;0}(\tilde{s})$$
$$zF_{m+1}(\tilde{s}, z) = \qquad (11)$$
$$\left(h_m(\tilde{s}) + h^\sharp_m(\tilde{s})\; z\right) F_m(\tilde{s}, z) - r_m(\tilde{s}) F_{m-1}(\tilde{s}, z)$$

This algorithm may be stated in an equivalent matrix presentation and the notation convention introduced in section 2 may ease the translation. This will be illustrated later for the final form of the test.

Stability Conditions

Define two auxiliary sequences of polynomials:

$$\varphi_m(\tilde{s}) := F_m(\tilde{s}, 1), \hat{\varphi}_m(\tilde{s}) := \frac{\varphi_m(\tilde{s})}{\varphi_0(\tilde{s})}, m = 0, ..., n \qquad (12)$$

Remark 2. It can be shown that all $\hat{\varphi}_m(\tilde{s})$ are actually polynomials. In other words, $\varphi_0(\tilde{s})$ is a factor of all $\varphi_m(\tilde{s})$ and $\hat{\varphi}_m(\tilde{s})$ is the result of dividing out this factor. It is also noted that (all $\varphi_m(\tilde{s})$ and therefore) all $\hat{\varphi}_m(\tilde{s})$ are (conjugate-) symmetric balanced polynomials, $\hat{\varphi}^\sharp_m = \hat{\varphi}_m$. A particular outcome of this symmetry is that all $\varphi_m(\tilde{s})$ are real $\forall s \in T$.

Theorem 2. (Stability Conditions for F - Table.) *Assume Algorithm 2 is applied to $D(z_1, z_2)$ and $\hat{\varphi}_m(\tilde{s})$ are formed as in (12). $D(z_1, z_2)$ is stable if, and only if, the following conditions (a) and (b) hold.*
(a) $D(z, 1) \neq 0$ *for all* $z \in \bar{V}$.
(b) $\hat{\varphi}_m(\tilde{s}) > 0$, $m = 1, \ldots, n$ *for all* $s \in T$.

Remark 3. It is possible to replace in the above theorem the condition (b) by the pair of conditions (i) $\hat{\varphi}_m(\tilde{s}) \neq 0$, $m = 1, \ldots, n$ for all $s \in T$ and (ii) $D(1, z)$ is a stable 1-D polynomial. Since the latter condition is necessary for stability, and is simple to check it adds a dividend that whenever $D(1, z)$ is determined to be not stable, the remaining 2-D test that contains the major portion of the computation burden may be skipped.

Remark 4. Condition (b) originates from the condition: **(b')** $Sgn\{\varphi_m(\tilde{s})\} = Sgn\{\varphi_0(\tilde{s})\} \; \forall s \in T$, $m = 1, \ldots, n$, and is equivalent to it. However (b') is much less convenient to use than (b) because $\varphi_0(\tilde{s})$ may have up to n zeros on T that are necessarily zeros of also all subsequent $\varphi_m(\tilde{s})$. Zeros on T are not uncommon (for odd n $\varphi_0(\tilde{s})$ has to have at least one zero at $s = -1$). As a consequence checking (b') would require numerical determination of the zeros of $\varphi_0(\tilde{s})$ on T and careful evaluation of the sign variations in segments of T partitioned by these zeros. In difference, condition (b) may be examined also by simple algebraic tests. In addition, the normalization (12 lowers the degrees of the polynomials to be tested that by itself affect positively the robustness and the cost of their testing.

Sharper Stability Conditions

The degree of each $\varphi_m(\tilde{s}) = F_m(\tilde{s}, 1)$ is equal to the row size of $F_m(\tilde{s}, z)$, denoted by $\ell_f(m)$. An expression for $\ell_f(m)$ may be obtained by solving the difference equation that it is seen from (12) to satisfy,

$$\ell_f(m+2) - 2\ell_f(m+1) - \ell_f(m) = 0 \, , \qquad (13)$$

for the initial conditions $\ell_f(-1) = \ell_f(0) = n_1$. The solution is a linear combination of two modes $\lambda_{1,2} = 1 \pm \sqrt{5}/2$. Its λ^m with $\lambda = 1 \pm \sqrt{5} (\approx 2.118)$ part causes the solution to increase exponentially with m. Consequently the positivity tests for $\varphi_m(\tilde{s}) \neq 0$ involves an amount of computation that increases rapidly with m. The next theorem facilitate this difficulty by establishing that one positivity test, that of $\varphi_n(\tilde{s}) > 0$, is enough.

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

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

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

(iii) $\hat{\varphi}_n(\tilde{s}) \neq 0$ *for all* $s \in T$

(iii') $\hat{\varphi}_n(\tilde{s}) > 0$ *for all* $s \in T$

5. Refined 2-D Stability Test

It turns out that recursion (12) generates a sequence of matrices $\{F_m\}$ that have row sizes higher than necessary. We first expose and characterize this phenomenon. Afterwards we devise another recursion that is free from this problem. Finally we shall obtain stability conditions for the reduced size table.

Redundant Factors

The next Lemma shows that each $F_m(\tilde{s}, z)$ $m \geq 2$ produced by the three-term recursion (12) contains separable polynomial in $\tilde{s}$ factors (i.e $F_m(\tilde{s}, z)$ is divisible by such factors with no reminder) that are inherited to subsequent $F_{m+i}(\tilde{s}, z)$ $i > 0$ and their degree increase rapidly.

Lemma 2. *Consider the sequence $\{F_m(\tilde{s}, z)\}_0^n$ produced by the recursion (12).*
(a) If $f(\tilde{s})$ is a factor of $F_m(\tilde{s}, z)$ $m \geq 0$ then it is a factor of all subsequent $F_{m+i}(\tilde{s}, z)$ $i \geq 1$.
(b) For any given two polynomials $G_0(\tilde{s}, z)$, $G_1(\tilde{s}, z)$ of degrees k and $k-1$ in z, respectively ($3 \leq k \leq n+1$), let $G_2(\tilde{s}, z)$, $G_3(\tilde{s}, z)$ be the two consecutive polynomials generated by the recursion (12) of Algorithm 2. Then $g_{[1]\ 0}(\tilde{s})g^{\sharp}_{[1]\ 0}(\tilde{s})$ is a factor of $G_3(\tilde{s}, z)$. Namely, $g_{[1]\ 0}(\tilde{s})g^{\sharp}_{[1]\ 0}(\tilde{s})$ divides, with no reminder, each $g_{[3]\ i}(\tilde{s})$ $i = 0, 1, \ldots, k-3$.

The conclusion from property (b) of Lemma 2 is that any $F_{k+2}(\tilde{s}, z)$, $k = 0, 1, \ldots$ is divisible by each of the factors $f_{[i]\ 0}(\tilde{s})f^{\sharp}_{[i]\ 0}(\tilde{s})$, $i = 0, \ldots, k$. By property (a), the multiplicity of each such factor tends to increase as the recursion goes on.

Construction

Lemma 2 provides sufficient details on the phenomenon it features to eliminate it. Only a recursive algorithm is capable to gain full advantage of the elimination of the common factors and create a most efficient 2-D table. Removing these factors by dividing them out after the F-table is completed does have a positive impact on testing stability conditions posed on the table, but it does not exploit as fully the phenomenon revealed in Lemma 2.

The next algorithm provides a recursion that associates $D(z_1, z_2)$ with a sequence $\{E_m(\tilde{s}, z) m = -1, 0, \ldots, n\}$ such that each $E_m(\tilde{s}, z)$ corresponds to $F_m(\tilde{s}, z)$ stripped from the aforementioned common factors.

Algorithm 3: Reduced 2-D table (E-table)
Construct for $D(z_1, z_2)$ a sequence of polynomials $\{E_m(\tilde{s}, z) = \sum_0^{n-m} e_k(\tilde{s})z^k,\ m = -1, 0, \ldots, n\}$ as follows.

Initiation. $E_{-1}(\tilde{s}, z) = (z-1)(\hat{D}(\tilde{s}, z) - \hat{D}^{\sharp}(\tilde{s}, z))$

$$E_0(\tilde{s}, z) = D(\tilde{s}, z) + D^{\sharp}(\tilde{s}, z) \quad (14)$$

Body. For $m = 0, 1, \ldots, n-1$ compute:

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

and let $q_{-1}(\tilde{s}) := 1$.

$$zE_{m+1}(\tilde{s}, z) = \frac{g_m(\tilde{s})E_m(\tilde{s}, z) + g^{\sharp}_m(\tilde{s})zE_m(\tilde{s}, z) - q_m(\tilde{s})E_{m-1}(\tilde{s}, z)}{q_{m-1}(\tilde{s})} \quad (15)$$

Stability Conditions

Define the auxiliary sequence of polynomials:

$$\epsilon_m(\tilde{s}) := E_m(\tilde{s}, 1),\ \hat{\epsilon}_m(\tilde{s}) := \frac{\epsilon_m(\tilde{s})}{\epsilon_0(\tilde{s})},\ m = 0, \ldots, n \quad (16)$$

They can be shown to hold properties similar to their counterparts in Remark 2.

The next stability theorem is the E-table parallel of Theorem 2.

Theorem 4. (Stability Condition for E-Table) *Consider $D(z_1, z_2)$ and apply to it Algorithm 3 and obtain $\hat{\epsilon}_m(\tilde{s})$'s of (16). $D(z_1, z_2)$ is stable if, and only if, the following conditions (a) and (b) hold.*

(a) $D(z, 1) \neq 0$ *for all* $z \in \bar{V}$.

(b) $\hat{\epsilon}_m(\tilde{s}) > 0$, $m = 1, 2, \ldots, n$ *for all* $s \in T$

Remark 5. Like in previous Remark 3, here too it is possible to replace in Theorem 4 condition (b) by the pair of conditions (i) $D(1, z) \neq 0$ for all $z \in \bar{V}$, (ii)$\hat{\epsilon}_m(\tilde{s}) \neq 0$, $m = 1, \ldots, n$ for all $s \in T$.

The next theorem crowns the E-table with the sharper set of stability conditions obtained before for the F-table. This proves at last that the factors dropped in the transition from the F-table to the E-table are truly redundant and their discard benefits the efficiency of computation without complicating the form of the stability conditions.

Theorem 5. (Sharp Stability Conditions for E-Table.) *$D(z_1, z_2)$ is stable if, and only if, the following three conditions: (i), (ii) and (iii) or (iii') hold.*

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

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

(iii) $\hat{\epsilon}_n(\tilde{s}) \neq 0$ *for all* $s \in T$

(iii') $\hat{\epsilon}_n(\tilde{s}) > 0$ *for all* $s \in T$

Comparison between the E-table and the F-table

The E-table has much lower row dimensionality than the F-table. For the F-table, $\ell_f(m)$, the degree in s of $F_m(s, z)$, is governed by (13) and increases exponentially with m. Let $\ell_e(m)$ denote the degree in s of $E_m(s, z)$ (i.e. the row size of E_m is $\ell_e(m) + 1$). Inspection on the recursion (15) reveals that for $m \geq 0$ it satisfies the equation $\ell_e(m+2) - 2\ell_e(m+1) + \ell_e(m) = 0$. Its solution for the initial values $\ell_e(0) = n_2$ and $\ell_e(1) = 3n_2$ is $\ell_e(m) = (2m+1)n_2$. Since $\ell_e(m)$ increase only linearly with m the E-table offers a substantial saving in the cost of computation by comparison to the F-table and this saving becomes more significant the higher are the values n_1 and n_2. The fact that we were able to show for the E-table stability conditions that look like those for the F-table means that the stability conditions for the E-table actually involve less computation because each $\hat{\epsilon}_m(s)$ is of lower degrees than $\hat{\varphi}_m(s)$. In particular, the highest degree symmetric polynomial that has to be tested $\hat{\epsilon}_n(s)$ is of degree $2n_1n_2$. This is according to Theorem 5 also the only polynomial that has to bested. The difference is remarkable already at low values of n_1 and n_2. For example consider $n_1 = n_2 = n = 4$. For the F-table $\{\ell_f(m)\}_0^4 = \{4, 12, 48, 68, 164\}$ and $\hat{\varphi}_n(s)$ is of degree 160. While for the E-table $\{\ell_e(m)\}_0^4 = \{4, 12, 20, 28, 36\}$ and $\hat{\epsilon}_n(s)$ is of degree 32.

Numerical Example

For illustration, consider the polynomial used as an example in several papers, [10] [7], $D(z_1, z_2) =$

$$[\begin{array}{ccc} 1 & z_1^1 & z_1^2 \end{array}] \left[\begin{array}{ccc} 0 & 0 & 0.2500 \\ 0 & 0.2500 & 0.5000 \\ 0.2500 & 0.5000 & 1.0000 \end{array}\right] \left[\begin{array}{c} 1 \\ z_2^1 \\ z_2^2 \end{array}\right]$$

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

$$E_{-1} = \left[\begin{array}{cccc} 1.0000 & -0.5000 & -0.5000 & 0 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0 & -0.5000 & -0.5000 & 1.0000 \end{array}\right]$$

$$E_0 = \left[\begin{array}{ccc} 1.0000 & 0.5000 & 0.5000 \\ 0.5000 & 0.5000 & 0.5000 \\ 0.5000 & 0.5000 & 1.0000 \end{array}\right]$$

$$E_1 = \left[\begin{array}{cc} 0.5000 & 0.5000 \\ 1.7500 & 1.5000 \\ 4.1250 & 3.7500 \\ 4.3750 & 4.3750 \\ 3.7500 & 4.1250 \\ 1.5000 & 1.7500 \\ 0.5000 & 0.5000 \end{array}\right]$$

The next step demonstrates the elimination of a redundant factor (since here $n_2 = 2$ this elimination occurs for the first time at this last step). Taking the first column of E_0, $e_{[0]\,0} = [1.000\,, 0.500\,, 0.500]^t$ we obtain $q_0 = e_{[0]\,0} * e_{[0]\,0}^{\sharp} = [0.5000\ , 0.7500\ , 1.5000\ , 0.7500\ , 0.5000]\mathbf{s}$. This polynomial divides all the $n_2 + 1 - 2$ column polynomials of the numerator matrix in the r.h.s of (15) which has in this example a single column that is given by: $[0.2500, 1.6875\ , 7.3750\ , 21.469\ , 48.102\ , 82.781\ , 115.02\ , 127.27\ , 115.02\ , 82.781\ , 48.102\ , 21.469\ , 7.3750\ , 1.6875\ , 0.2500]^t$. The result of the deconvolution by q_0 is:

$$E_2 = \left[\begin{array}{c} 0.5000 \\ 2.6250 \\ 9.3125 \\ 20.344 \\ 33.312 \\ 37.969 \\ 33.312 \\ 20.344 \\ 9.3125 \\ 2.6250 \\ 0.5000 \end{array}\right]$$

Next $\hat{\epsilon}_2(s) = E_2'\mathbf{s}/\epsilon_0(s)$ is to be determined, where $\epsilon_0(s) = E_0(s,1) = [2.000\,, 1.500\,, 2.000]\mathbf{s}$. The result is $\hat{\epsilon}_2(s) = [0.2500, 1.1250, 3.5625, 6.3750, 8.3125, 6.375, 3.5625, 1.1250, 0.2500]\mathbf{s}$. It remains to test the condition $\hat{\epsilon}_2(s) \neq 0\ \forall s \in T$. An efficient way to do so is to use the 1-D zero location test for real polynomials in [12]. Finding that this condition holds ends the 2-D stability test with the conclusion that the examined $D(z_1, z_2)$ is stable.

REFERENCES

[1] J. L. Shanks, S. Treitel, and J. H. Justice, "Stability and synthesis of two-dimensional recursive filters. *IEEE Trans. Audio Electroacoustics*, Vol. AU-20, pp. 115-128, June 1972.

[2] N. K. Bose *Applied Multidimensional System Theory* (Chapter 3), 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" *Multidimensional Systems: Techniques and Applications*, S. G. Tzafestas Ed., New York:Marcel Dekker,1986, Ch. 3.

[5] Y. Bistritz, "A Modified Unit Circle Zero Location Test" to appear in *IEEE Trans. Circuits and Systems.*

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

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

[8] E. I. Jury "A Note on the Modified Stability Table for Linear Discrete Time System", *IEEE Trans. on Circuits and Systems.* vol. CAS-38, pp. 221-223, 1991.

[9] Y. Bistritz, "Zero clustering in the unit bi-circle of two-variable polynomials", submitted for publication.

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

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

[12] 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.

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