IEEE International Symposium on Circuits and Systems, ISCAS '98, Monterey, CA, May 1998

A STABILITY TEST OF REDUCED COMPLEXITY FOR 2-D DIGITAL SYSTEM POLYNOMIAL

Yuval Bistritz

Department of Electrical Engineering
Tel Aviv University, Tel Aviv 69978, Israel
bistritz@eng.tau.ac.il

ABSTRACT

A new algebraic test for deciding whether a 2-D (bivariate) polynomial has no zeros in the closed exterior of the unit bi-circle is presented. The testing of a polynomial of degree (n_1, n_2) is performed by $n_1 n_2 + 2$ unit circle tests of 1-D polynomials of degree n_1 or n_2 plus one of degree $2n_1 n_2$ and it can be carried out in a very low (apparently unprecedented) count of approximately $1.5 n_1 n_2^3 + 2n_1^2 n_2^2$ real flops.

1. 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

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

are used to denote the unit circle, its interior, and its exterior, respectively, and the bar denotes closure, e.g. $\bar{V} = V \cup T$. A stable 2-D polynomial is the key for the stability of 2-D linear shift-invariant (discrete,digital) recursive filters and systems [1],[2]. Similarly, a 1-D polynomial $P(z)$ that has all its zeros in U will be called a (1-D) stable polynomial.

The current paper presents a new computationally efficient approach to testing 2-D stability conditions for of $D(z_1, z_2)$ proposed first in [3]. The test proposed there constructs for $D(z_1, z_2)$ a sequence of matrices $\{E_m, m = 0, \ldots, n\}$ and poses on it stability condition. This approach may be called a 'tabular' 2-D stability test Calling this sequence of matrices 2-D 'stability table' considers them the 2-D extension of the tradition in 1-D stability test tables, with rows of the table in the 1-D case being replaced by matrices. These rows/matrices for resting 1-D/2-D stability may alternatively be presented and and more effectively managed as a sequence of 1-D/2-D polynomials. The main result from [3] that will needed for the current derivation is repeated briefly as the next algorithm and theorem. The notation that is used in the following associates 2-D polynomials with their coefficient matrix: $E_m(\tilde{s}, z) = \tilde{\mathbf{s}}^t E_m \mathbf{z}$ where $\mathbf{z} := [1, z, \ldots, z^i, \ldots]^t$ (of appropriate length) and where $\tilde{\mathbf{s}} := [s^{-n}, \ldots, s^{-1}, 1, s, \ldots, s^n]^t$ for certain integers n (all the matrices in the sequence have odd number of rows). This notations may also associate vectors and 1-D polynomials. We also denote $E_m^\sharp := J E_m J$ for a matrix and $e_k^\sharp := J e_k$ for a vector, where J is the reversion matrix of appropriate size.

Algorithm 1: 2-D stability table. Construct for $D(z_1, z_2)$ the sequence of polynomials.

$$E_m(\tilde{s}, z) = \sum_{k=0}^{n-m} e_{[m]\ k}(\tilde{s}) z^k, \quad , \quad m = 0, 1, \ldots, n\ (= n_2)$$

(i) Initiation. $M(\tilde{s}, z) = D(s^{-1}, 1) D(s, z)$

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

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

$$q_0(\tilde{s}) = E_0(\tilde{s}, 1)$$

(ii) Recursion. For $m = 1, \ldots, n-1$ obtain $E_{m+1}(\tilde{s}, z)$:

$$g_m(\tilde{s}) = e_{[m-1]\ 0}(\tilde{s}) e^\sharp_{[m]\ 0}(\tilde{s})$$

$$q_m(\tilde{s}) = e_{[m]\ 0}(\tilde{s}) e^\sharp_{[m]\ 0}(\tilde{s})$$

$$z E_{m+1}(\tilde{s}, z) =$$

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

The use of $\tilde{\mathbf{s}}$ instead of $\mathbf{s}^t E_m \mathbf{z}$ with $\mathbf{s} = [1, s, \ldots, s^i, \ldots]^t$, was of advantage during the original derivation of the algorithm and will be helpful also in the forthcoming derivation of link 1-D stability tests when $s \in T$. Otherwise either convention may be used, or both may be dropped in favor of regarding the algorithm as operating on vectors and matrices where multiplication / division between a vector and a matrix mean convolution / deconvolution between the vector and each column of the matrix. The divisions by $q_m(\tilde{s})$ represent true elimination of common polynomial factors in the numerator. This elimination reduces the row sizes of the E_m matrices and improves the efficiency of the algorithm. The matrices E_m produced by the algorithm are all centro-symmetric real matrices, $E_m^\sharp = E$. It therefore suffices to calculate only half of their entries.

Theorem 1. (Stability Conditions for Algorithm 1.) *$D(z_1, z_2)$ is stable if, and only if, the following three conditions: (i), (ii) and (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) $\epsilon(s) := \mathbf{s}^t E_n \neq 0$ *for all* $s \in T$

The first two conditions involve two 1-D stability tests. The polynomial $\epsilon(s)$ in condition (iii) is determined by the last array E_n of the 2-D table produced by Algorithm 1. The size of E_n $(2n_1n_2 + 1) \times 1$ and centro-symmetry implies it is a symmetric vector. So, $J\epsilon = \epsilon$ and $\epsilon(s)$ is a symmetric polynomial of degree $2n_1n_2$.

This 2-D stability test is of order n^6 (say $n_1 = n_2 = n$) complexity of an efficiency that compares well with previous 2-D tabular tests.

An acute question that we raise currently is whether it is possible to bring forth the last polynomial of the above 2-D table, $\epsilon(s) = \mathbf{s}^t E_n$, in a computationally less consuming manner. The question is motivated by the fact that in view of the stability conditions in Theorem 1, the sole purpose of the production of the whole sequence $\{E_m, m = 0, 1, \ldots, n\}$ is to obtain its last component. Anticipation for a more efficient solution stems from the following considerations. A polynomial of degree N can be completely determined from its value at $N+1$ points. In general, such interpolation of a polynomial from $N+1$ distinct values leads to a Vandermonde set of equations that can be solved in order N^2 of operations (cf. [4] p. 121). Thus, the interpolation of $\epsilon(s)$ of condition (iii) from $2n_1n_2+1$ values may be expected to have an order n^4 solution (taking $n_1 = n_2 = n$ for the current argument). As was noted already in [3], condition (iii) can be tested in $n_1^2 n_2^2$ arithmetic operations using the method in [5]. So, an order n^4 solution to the problem should be possible if each interpolated value can be obtained in order n^2 of operations which we proceed to show is feasible. The actual final cost of computation will depend on how the interpolation points are chosen, how the interpolated values are obtained and how the interpolation problem is solved.

2. COMPANION 1-D STABILITY TEST

It follows from the manner that the 2-D stability table was originally derived [3] that it it is possible to determine values b_i of $\epsilon(\tilde{s})$ at desirable $s_i \in T$, $b_i = \epsilon(\tilde{s}_i)$, by tracing the effect of algorithm 2 on the 1-D polynomial (with complex coefficients!) $P_{s_i}(z) = D(s_i, z)$. The algorithm below represents a 1-D projection of the 2-D polynomial algorithm 1. (* denotes complex conjugate.)

Algorithm 2. Assume $P(z)$ is a polynomial of degree n with complex coefficients and that $P(1) \neq 0$. Form $\hat{P}(z) = P(1)^\star P(z)$ and construct the next sequence $\{E_m(z), m = 0, 1, \ldots n\}$, $E_m(z) = \sum_{i=0}^{n-m} e_{m,i} z^i$ of (conjugate) symmetric polynomials (i.e. $JE_m = E_m^\star$).

(i) Initiation. $E_0(z) = \hat{P}(z) + \hat{P}^\sharp(z)$

$$E_1(z) = \frac{\hat{P}(z) - \hat{P}^\sharp(z)}{(z-1)} \quad , \quad q_0 = 2|P(1)|^2$$

(ii) Recursion. For $m = 1, \ldots, n-1$ obtain $E_{m+1}(z)$:

$$g_m = e_{m-1,0} e_{m,0}^\star \quad , \quad q_m = |e_{m,0}|^2$$

$$zE_{m+1} = \frac{(g_m + g_m^\star z) E_m(z) - q_m E_{m-1}(z)}{q_{m-1}}$$

The requirement in the algorithm that $P(1) \neq 0$ will be satisfied in its forthcoming applications. The above recursions are written in a form that leaves their relation to the 2-D recursion of Algorithm 2 transparent. A more effective manner for computation is to carry it out with just two multipliers per recursion step, g_m/q_{m-1} and q_m/q_{m-1}. This is possible now because g_m and q_m become scalars.

The next theorem attaches stability conditions to Algorithm 2 such that together they become a rightful 1-D stability test for $P(z)$.

Theorem 2. (Stability conditions for Algorithm 2.) **(a)** *$P(z)$ is stable, if and only if, $E_m(1) > 0$, $m = 0, 1, \ldots, n$. where $\{E_m(z)\}$ are obtained by Algorithm 2.* **(b)** *If $e_{m,0} = 0$ then $P(z)$ is not stable.*

The proof follows from the relation of Algorithm 1 to the stability test for complex polynomials in [6]. The details will be made available elsewhere. Note that the stability of any 1-D polynomials $P_{s_i}(z) = D(s_i, z)$, $s_i \in T$, is necessary condition for 2-D stability of $D(z_1, z_2)$. Consequently this theorem will be a useful enhancement to obtaining interpolation points by Algorithm 2. The incorporation of its necessary conditions for stability into the execution of Algorithm 2 may save computation by interruption of the rest of the test as soon as violation of any necessary condition for stability is detected.

3. THE INTERPOLATION PROBLEM

We want to determine the vector ϵ from known values $b_i = \epsilon(\tilde{s}_i)$ that will be provided by the application of Algorithm 2 to $P_{\tilde{s}_i}(z) := D(s_i, z)$ at desirable interpolation points $s_i \in T$. A collecting of $2M+1$ values of $\epsilon(\tilde{s}_i)$ at distinct points produces the next set of equations. For $i = 0, 1, \ldots, 2M$:

$$[s_i^{-M}, s_i^{-M+1}, \ldots, s_i^{-1}, 1, s_i, \ldots, s_i^{M-1}, s_i^{M}]\epsilon = b_i$$

This set may be solved and provide the required vector ϵ. It is of advantage to choose $2M$ of the interpolation points in conjugate pairs, s_i and $s_{i+1} = s_i^{-1}$ because than $b_{i+1} = b_i$. Therefore it will suffice to run Algorithm 2 only $M+1$ instead of $2M+1$ times. It is possible to choose the interpolation points equally spaced on T in consistency with this benefit and add a DFT-like orthogonality property to the a set of equations that alleviates the matrix inversion. The details of the proposed solution are as follows. Define $w = e^{j\theta}$, $\theta = \frac{2\pi}{2M+1}$ where ($j = \sqrt{-1}$). Then choose $s_{M+i} = w^{-i} \quad i = 0, \pm 1, \pm 2, \ldots, \pm M$. The set of equations to be solved for $\epsilon = [\epsilon_0 \ldots \epsilon_{2M}]^t$ becomes $Q\epsilon = b$ where $b = [b_0, \ldots, b_{2M}]^t$ are the known interpolation values and Q takes the form:

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

The rows r_k^t and columns c_k of Q are of the form $[w^{-Mk}\ , w^{-(M-1)k},\ \ldots, w^{-k}, 1,\ w^{k}, \ldots,\ w^{(M-1)k}, w^{Mk}]^t$ appearing in the order $k = -M, \ldots, -1, 0, 1, \ldots, M$. The inner product of two such vectors reveals the following orthogonality.

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

It follows that

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

Therefore the sought vector gets the explicit expression $\epsilon = \frac{1}{2M+1} Qb$, after also using the symmetry of $b = Jb$. (In the following we shall drop the $2M+1$ scaling factor.) The symmetries of ϵ and b permit further simplifications. Since ϵ is symmetric, it is enough to read only the first $M+1$ rows of this explicit solution. The symmetry of b may be used to 'fold' it to half length while moving to cosine terms. Namely, pair of terms $w^{(M-k)(M-i)} b_i +\ w^{-(M-k)(M-i)} b_{2M-i} = 2b_i \cos((M-k)(M-i)\theta)$ may be collected for $i = 0, \ldots, M-1$. Finally, the coefficient vector of $\epsilon(s)$ can be obtained from the interpolation values b_i , $i = 0, 1, \ldots, M$ by the next simple expression.

$$\begin{aligned} \epsilon_{M-m} &= b_M + 2\sum_{k=1}^{M} b_{M-k} \cos(mk\theta)\,, \ m = 0, \ldots, M \\ \epsilon_{M+m} &= \epsilon_{M-m} \qquad , \quad m = 1, \ldots, M \end{aligned} \tag{3}$$

4. THE NEW 2-D STABILITY TEST

The method that emerges from the previous results for testing whether $D(z_1, z_2)$ is 2-D stable (whether (2) is true for (1)) is summarized as the next 4 steps.

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

Step 2. Let $M = n_1 n_2$, $\theta = \frac{2\pi}{2M+1}$, $w = e^{j\theta}(j = \sqrt{-1})$. For $m = 0, 1 \ldots, M$ do:

Set $s_m = w^m$. Apply to $P_{s_m}(z) = D(s_m, z)$ Algorithm 2 and check the accompanying 1-D stability conditions of Theorem 2. If not 1-D stable - 'exit'. Otherwise, retain $E_n (> 0)$ as $b_{M-m} := E_n$.

Step 3. Calculate the coefficients of the polynomial $\epsilon(s) = \sum_{i=0}^{2M+1} \epsilon_m s^i$ from the values $b_m (> 0)$, $m = 0, \ldots, M$ by using (3).

Step 4. Examine the condition $\epsilon(s) \neq 0$ for all $s \in T$. $D(z_1, z_2)$ is stable if and only if this condition holds and the current step has been reached without an earlier 'exit'.

Remarks:
1. The condition in Step 4 may examined in several numerical or algebraic ways. It may be useful to this end to recall from [3] that this condition (iii) of Theorem 1 is replaceable by the positivity condition $\epsilon(\tilde{s}) > 0$ for all $s \in T$ (where replaceable means that in the context of the theorem, the two conditions are equivalent). This positivity condition may also be arranged as
$\epsilon(e^{j\theta}) = \epsilon_M + 2\sum_{k=0}^{M-1} \epsilon_k \cos((M-k)\theta) > 0 \, \forall\theta$
Algebraic means to examine this condition include translation of the cosines via Chebyshev polynomials to a polynomial of degree M and testing it for no zeros on [-1,1] by a Sturm sequence. The algebraic method of least count of operations for step 4, that examines $\epsilon(s) \neq 0$ for all $s \in T$ directly, is the zero location test in [5]. We assume this choice in the count of arithmetic operations carried out below.
2. Note that condition (ii) of Theorem 1 is carried out step 2 at $m = 0$.
3. If the condition in step 1 is true than all $P_{s_m}(z)$ in step 2 conform with the requirement $P_{s_m}(1) \neq 0$ in Algorithm 2.

5. COUNT OF OPERATIONS

The presented procedure may be carried out in $1.5n_1n_2^3 + 2n_1^2n_2^2 + O(n_{1,2}^3)$ multiplications and $1.5n_1n_2^3 + 3n_1^2n_2^2 + O(n_{1,2}^3)$ additions where here and after $O(n_{1,2}^3)$ is used to mean dropping terms $n_1^{\alpha_1}n_2^{\alpha_2}$ such that $\alpha_1 + \alpha_2 \leq 3$. This count assumes that the test is completed by purely algebraic means and specifically that the zero location test of [5] is used for step 4. These figures are obtained as follows.

Step 1 is a 1-D stability test for a real polynomial of degree n_1. It can be carried out in a negligible order n_1^2 count of operations by most available 1-D stability tests (including that of §2). The method in [5], that is expected to be made available for step 4, is also the method of least count of operations to test step 1, requiring for it $0.25n_1^2 + O(n_1)$ multiplications. Step 2 involves n_1n_2 tests of complex 1-D polynomials (and the real one $P_{s_0}(s)$) each of degree n_2. Algorithm 2 requires $1.5n^2 + O(n)$ real multiplications and additions. This calculation counts multiplication of two complex numbers as 4 real multiplications and 2 additions, a real times complex numbers as 2 real multiplications, and assumes the symmetry of the polynomials is exploited to compute only half of the coefficients. Thus, Step 2 requires $1.5n_1n_2^3 + O(n_{1,2}^3)$ real multiplications and additions. Step 3 requires $n_1^2n_2^2 + O(n_{1,2}^3)$ real multiplications and additions. The most efficient algebraic method for testing the condition in Step 4, the method in [5] requires $0.25n^2 + O(n)$ multiplications and $0.5n^2 + O(n)$ additions for a polynomial of degree n. Thus, step 4 may be performed in $n_1^2n_2^2$ multiplications and $2n_1^2n_2^2$ additions + $O(n_{1,2}^3)$. The summation of these counts yields the cost estimate at the opening of this section.

6. CONCLUSION

A new efficient test to decide whether all the zeros of a two-variable polynomial reside in the interior of the unit bi-circle has been presented. To the author's best knowledge, the efficiency (measured by counts of real flops) of the test established in this paper exceeds any other method available in the literature for the same task. The test stems from the 2-D stability test proposed first in [3] that involves construction of a 2-D table and inspection of its last array. The low level of cost of computation is achieved by 'telepolation' - telescoping by interpolation - of the required last array of the 2-D table without its actual construction. A conceptually attractive feature of this telepolation approach is that the 2-D unit bi-circle stability test consists of $2n_1n_2 + 2$ low degree 1-D unit-circle stability tests plus one unit circle zero location test of degree $2n_1n_2$. This feature is also of practical value, because the 1-D stability tests pose on the telepolation process many necessary conditions for 2-D stability that may be used to abort the rest of the test with the first necessary condition for stability found not to hold.

7. REFERENCES

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

[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 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.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins U. Press, 1983.

[5] 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.

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