An Immittance-Type Stability Test for 2-D Digital Filters

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

Abstract

The paper presents a 'tabular' stability test for 2-D linear shift invariant systems that is based on the stability test for 1-D discrete-time system polynomials introduced by Bistritz. The algorithm involves simple manipulations of 1-D and 2-D polynomial functions or corresponding and equivalent manipulations of vectors and matrices. The total computation cost is of order n^6 (for a polynomial with variables of degrees are $n1 = n2 = n$) compared to typical costs that grow exponentially with n for previously reported 2-D tabular tests. The set of necessary and sufficient conditions for 2-D stability involves testing stability of a 1-D polynomial of degree n and testing positivity on an interval of a polynomial of degree $2n^2$.

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. $\bar{V} = V \cup T$.

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

This paper proposes a new 2-D tabular scheme to test whether or not a given 2-D polynomial is stable. The stability test is based on the efficient 1-D test for real polynomials in [2] and it is algebraic, namely, it determines whether $D(z_1, z_2)$ is stable or not in a finite number of arithmetic operation. The 2-D table consists of a sequence of matrices - the "2-D table" - and stability conditions posed on it. The construction of the table is governed by an generalization of the recursion for 1-D polynomials in [2] to a three-term recursion of 2-D polynomials. The adjective *immittance* in the title has become attached to this 1-D stability method following the extension of the approach to related signal processing applications [3], [4].

A previous 2-D stability test proposed in [10] was also based on the underlying 1-D test. It used a slightly different form of it [5], that differs from [2] in details that makes it closer in look to the Routh stability test for continuous-time system. Since The form in [2] is more pleasant for stability testing and is preferable for most applications when the similarity to the analogue test is not required.

We start with the 1-D test in its form in [2] and first modify it so as to avoid the arithmetic operation of division. From this modified form we obtain a preliminary 2-D stability testing procedure. This test already has the advantages over the implementation in [10] in manipulation and testing of polynomials rather than rational functions. Afterwards, a series of refinements are introduced to further simplify the construction of the 2-D table and to lessen the number of stability conditions posed on it. This set of stability conditions may be reduced to one 1-D stability test and one test for positivity of a polynomial on T. A similar type of simplification was first introduced to 2-D stability testing by Siljak who showed in [12] 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. The simplification was adopted for the 2-D tabular test in [8] using the relation of the modified Jury table there to the Schur-Cohn matrix determinants.

1058-6393/95 $4.00 © 1995 IEEE

The structure of our initial 2-D table is in itself further simplified by removing from it common factors that are shown to be redundant. Starting from an initial exponential order complexity (typical to also other earlier 2-D table tests [11]), the overall cost of computation for the reduced table with the simplified stability condition is of order n^6 (for $n = n_1 = n_2$).

1.1 Background Theory and Notation

The matrix of coefficients of a $D(z_1, z_2)$ such as (1) will be denoted by $D = (d_{i,k})$. Similarly for a 1-D polynomials, say $P(z)$, p will denote the vector of its coefficients. 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

$$D(z_1, z_2) = \sum_{k=0}^{n} d_k(z_1) z_2^k =$$

$$[d_0(z_1), d_1(z_1), \ldots, d_n(z_1)]\mathbf{z_2} = \mathbf{z_1}^t D \mathbf{z_2} \quad .$$

Here d_k is the $k+1$-th column of D, and $d_k(z_1) = \mathbf{z_1}^t d_k$, $k = 0, \ldots, n$, are the coefficient polynomials of $D(z_1, z_2)$ regarded as a 1-D polynomial in the second variable z_2. The latter view is also the standard approach to testing 2-D stability. The 2-D polynomial is regarded as a 1-D polynomial in one of its variables whose coefficients depend on the other variable. Then a 1-D stability test is applied in conjunction with an appropriate simplification of the stability condition in (2). We shall use the simplification stated in the next Lemma. Several other simplifying conditions of this kind are also known [9].

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

(i) $D(z, a) \neq 0$ for all $z \in \bar{V}$ 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 [13]. It is often called Huang's theorem who obtained it first in [7] for $a = \infty$. Note that part (a) is a simple 1-D polynomial stability test so that the task of an algebraic 2-D stability test concerns essentially an efficient way to test the condition (ii).

We shall form for the tested polynomial a sequences of bivariate polynomials of the form $\{F_m(x, z), 0, \ldots, n\}$, or equivalently , a sequence of matrices $\{F_m , 0, \ldots, n\}$ defined by their coefficients (that will also be called the 2-D 'table'). The index m will be added to the coefficient matrix entries in bracket preceding the matrix indices, $F_m = (f_{[m]i,k})$. Thus,

$$F_m(x, z) = [f_{[m]\ 0}(x) \ , \ f_{[m]\ 1}(x) \ , \ldots]\mathbf{z} =$$

$$\mathbf{x}^t [f_{[m]\ 0} \ , \ f_{[m]\ 1} \ , \ \ldots]\mathbf{z}$$

The usual way to bring the multidimensional condition (ii) in Lemma 1 to comply with a 1-D test for *real* polynomials is to regard $D(s, z)$ as a polynomial in z with coefficients in $s \in T$ and multiply it by the polynomial formed by complex conjugation of its coefficients [6] [10]. Define for the polynomial $D(z_1, z_2)$, the polynomial

$$Q(s^{-1}, s, z) = D(s^{-1}, z) D(s, z) = \tilde{\mathbf{s}}^t Q \mathbf{z} \quad , \qquad (3)$$

where $\tilde{\mathbf{s}} := [s^{-n_1}, \ldots, s^{-1}, 1, s, \ldots s^{n_1}]^t$. This operation creates a matrix Q of size $(2n_1+1) \times (2n_2+1)$. The columns q_i of Q are symmetric vectors, i.e. $Jq_i = q_i$ where J is the reversion matrix (with 1's on the antidiagonal and zeros elsewhere). By mapping the coefficient polynomials from T to the finite interval $[-1, 1]$ via the transformation

$$x = \frac{s + s^{-1}}{2} \quad s \in T \quad , \quad x \in [-1, 1] \qquad (4)$$

it is possible to return to the original row size. We shall denote the resulting 2-D polynomial by

$$R(x, z) = Q(s^{-1}, s, z)_{|x = \frac{1}{2}(s + s^{-1})} \qquad (5)$$

The substitution needed in (5) may be obtained by exploiting trigonometric relations that follows from regarding s and x as $s = e^{j\theta}$ and $x = cos\theta$ ($j = \sqrt{-1}$), as shown in [6]. This is also equivalent to replacing a series expansion in Chebyshev polynomials $T_m(x) = \frac{1}{2}[s^m + s^{-m}]$ by expansion in a power series . A MATLAB routine for this and one that performs the conversion from the matrix D to the matrix R, where $D(z_1, z_2) = \mathbf{z_1}^t D \mathbf{z_2}$ and $R(x, z) = \mathbf{x}^t R \mathbf{z}$ is given in an appendix of [1].

It is easily seen that the testing of condition (ii) in the context of Lemma 1 is equivalent to testing $R(x, z)$ for the condition

$$R(x, z) \neq 0 \quad \forall x \in [-1, 1] \quad \text{and} \quad \forall z \in \bar{V} \qquad (6)$$

1.2 The Underlying 1-D Test

A polynomial $P(z)$ of degree n is called stable if all its n zeros reside in U, or equivalently if

$$P(z) = \sum_{i=0}^{n} p_i z^i \neq 0 \quad \forall z \in \bar{V} \quad . \qquad (7)$$

An efficient way to test the stability of a real polynomial $P(z)$ was proposed in [2]. The algorithm in [2] uses a recursion that contains one division operation per recursion step. Its generalization to test the condition (ii) in Lemma 1 would lead to an algorithm that propagates and examines rational functions. In order to circumvent this undesirable complication we shall use the following modified 1-D test.

It is relatively simple to establish, using the results in [2], the following modified algorithm and theorem [1].

Algorithm 1 : Division-Free 1-D Table. Given $P(z)$ and assume $P(1) > 0$ (or $p_n > 0$) assign to it a sequence of $n+1$ symmetric polynomials $\{F_m(z)\}_0^n$

$$F_m(z) = \sum_{i=0}^{n-m} f_{m,i} z^i, \text{ (symmetry} \mapsto f_{m,n-m-i} = f_{m,i} \text{)}$$

as follows:

(i) **Initiation.**

$$F_0(z) = P(z) + z^n P(z^{-1}), \; F_1(z) = \frac{P(z) - z^n P(z^{-1})}{(z-1)}$$

(ii) **Recursion.** For $m = 0, \ldots, n-2$:

$$zF_{m+2}(z) = f_{m,0}(z+1)F_{m+1}(z) - f_{m+1,0}F_m(z) \quad (8)$$

Theorem 1. (Stability Conditions for Modified 1-D Test). *The polynomial $P(z)$ is stable if and only if Algorithm 1 yields for it*

$$\varphi_m := F_m(1) > 0 \qquad m = 0, 1, \ldots, n \quad . \quad (9)$$

2 The First 2-D Test

This section brings a first form of a 2-D table that is based on using the previous modified 1-D test to implement the condition (6). Afterwards an accompanying first set of necessary and sufficient stability conditions are first provided and then simplified.

2.1 First 2-D Table

The following algorithm assigns to $R(x, z)$ of (5) a sequence of $n+1$ 2-D polynomials ($n := 2n_2$) $\{F_m(x, z)\}_0^n$

$$F_m(x, z) = \sum_{k=0}^{n-m} f_{[m]\ k}(x) z^k \,, m = 0, \ldots, n \qquad (10)$$

This sequence, or more specifically, the sequence of matrix arrays $\{F_m = (f_{[m]i,k})\}_0^n$ will be refered the initial 2-D table or "F-table". The algorithm is a direct application of Algorithm 1 to $R(x, z)$ regarding it as a 1-D polynomial in the variable z of degree $n := 2n_2$ with coefficients that are 1-D polynomial in x, viz., $R(x, z) = \sum_0^n r_i(x) z^i$.

Algorithm 2: The "F-Table".

For $R(x, z)$ that is obtained for $D(z_1, z_2)$ via and (3)-(5) construct $\{F_m(x, z), m = 0, 1, ..., n\}$ as follows

(i) **Initiation.** $F_0(x, z) = R(x, z) + z^n R(x, z^{-1})$

$$F_1(x, z) = \frac{R(x, z) - z^n R(x, z^{-1})}{z - 1}$$

(ii)**Recursion.** For $m = 0, 1, ..., n-2$, $zF_{m+2}(x, z) =$

$$f_{[m]\ 0}(x)(z+1)F_{m+1}(x, z) - f_{[m+1]\ 0}(x)F_m(x, z) \quad (11)$$

The symmetry of the polynomials in Algorithm 1 induces a symmetry on the columns of F_m

$$f_{[m]\ k}(x) = f_{[m]\ n-m-k}(x) \,, \; k = 0, \ldots, n-m$$

This symmetry may be used to actually compute only half of the columns of each F_m.

The next stability conditions for Algorithm 2 may be proved quite straightforwardly from a combination of Lemma 1 with Theorem 1.

Theorem 2. (Stability conditions for the F-table). *Assume $R(x, z)$ corresponds to $D(z_1, z_2)$ through (3)-(5) and that Algorithm 2 produced for it the sequence $\{F_m(x, z)\}_0^n$. The following conditions (i)-(iii) form a set of necessary and sufficient conditions for $D(z_1, z_2)$ to be stable:*

(i) $D(z_1, 1) \neq 0 \qquad \forall z_1 \in \bar{V}$

(ii)$D(1, z_2) \neq 0 \qquad \forall z_2 \in \bar{V}$

(iii) $\varphi_m(x) \neq 0 \, \forall x \in [-1, 1] \; m = 0, 1, \ldots, n$

where $\varphi_m(x) := F_m(x, 1)$

From a mathematical point of view, it is possible to drop in Theorem 2 condition (ii), and change the condition (iii) to read $\varphi_m(x) > 0$ for all m. It is our view however that the computation involved in an extra 1-D test is negligible by comparison to the total computational cost and the testing of whether $\varphi_m(x) \neq 0$ is sometimes simpler than testing $\varphi_m(x) > 0$. In addition testing simple extra necessary condition like this 1-D stability test (and maybe additional convenient 1-D tests, like $D(-1, z)$, $D(z, -1)$, $D(0, z)$, $D(z, 0)$ and $D(z, z)$) may be rewarding by saving at times the more laborious computation requested for the rest of the 2-D test.

2.2 Refined Conditions for the F-Table

This subsection asserts that it is in fact sufficient for stability determination to examine positivity over (or even just no zeros in) $[-1, 1]$ for only the last entry of the 2-D table.

Theorem 3. (Refined conditions for the F-Table). *Assume $R(x, z)$ is obtained for $D(z_1, z_2)$ through (3)-(5) and that Algorithm 2 applied to it produced $\{F_m(x, z)\}$. The following conditions (i)-(iii) form a set of necessary and sufficient conditions for $D(z_1, z_2)$ to be stable:*
(i) $D(z_1, 1) \neq 0 \quad \forall z_1 \in \bar{V}$
(ii) $D(1, z_2) \neq 0 \quad \forall z_2 \in \bar{V}$
(iii) $\varphi_n(x) := F_n(x, 1) \neq 0 \quad \forall x \in [-1, 1]$

The proof of this theorem involves a judicious examination of the structure of Algorithm 2. It will become available in the full version form of this conference abridged presentation [1].

The testing of the condition $\varphi_n(x) \neq 0$ (to which we refer for brevity a 'positivity tests') can be done in several ways. One algebraic approach is to build a Sturm sequence for $\varphi_n(x)$ and testing sign variation conditions at the end points of the interval $[-1, +1]$. A second approach to be detailed in [1] is to use the "type I" singularity case in [2]. The latter approach requires only a slight extra programming because it is part of the zero location method from which the current 2-D stability test is derived.

3 The Reduced 2-D Stability Test

It turns out that Algorithm 2 creates a sequence of matrices $\{F_m\}$ whose row dimensionality is (much) higher than necessary. After quantifying this observation we propose a revised recursion that produces a sequence $\{E_m(x, z)\}$ of 2-D polynomials devoid of this redundancy. Finally the section brings necessary and sufficient conditions for stability for the reduced 2-D table $\{E_m(x, z)\}$.

3.1 Redundant Common Factors

The next lemma reveals and features a pattern of accumulation of common $x-$polynomial factors in the sequence $\{F_m(x, z)\}$.

Lemma 2. *Consider the sequence $\{F_m(x, z)\}_0^n$ produced by the recursion (11).*
(a) If $f(x)$ is a factor of $F_m(x, z)$ $m \geq 1$ then it is a factor of all subsequent $F_{m+i}(x, z)$ $i \geq 1$.
(b) Let any two adjacent polynomials $G_0(x, z) = F_m(x, z)$ and $G_1(x, z) = F_{m+1}(x, z)$ of degrees k and $k-1$, $4 \leq k \leq n-3$, in the z variable, and let three immediate consecutive polynomials generated by the above recursion be $G_2(x, z)$, $G_3(x, z)$ and $G_4(x, z)$ (of degrees $k-2$ $k-3$ and $k-4$ in z, respectively). Then $g_{[1]\ 0}(x)$ is a factor common to all the coefficients $g_{[4]\ i}(x)$ $i = 0, 1, \ldots, k-4$ of $G_4(x, z)$ (viewed as a polynomial in z). In other words, the polynomial $g_{[1]\ 0}(x)$ divides $G_4(x, z)$ with no reminder.

The proof of this Lemma is omitted and will become available in [1]

3.2 Reduced 2-D table

The last lemma claims that each polynomial $F_{3+k}(x, z)$ in Algorithm 2, $k \geq 1$ contains a build-up of the following factors: $f_{[i]\ 0}(x)$ $\quad i = 1, \ldots, k$. The structure of the recursion admits the elimination of these repetitive factors recursively as soon as they occur. It will be shown subsequently that these are redundant factors and that equally simple stability conditions applies for the resulting reduced table form.

The next algorithm provides a recursion that produces a sequence of n 2-D polynomials $E_m(x, z)$ such that for each m $E_m(x, z)$ corresponds to $F_m(x, z)$ after dividing out from it all the common factors $f_{[i]\ 0}(x)$ $\quad i \leq m$.

Algorithm 3: The 2-D "E-Table"

Consider $R(x, z)$ that has been obtained for $D(z_1, z_2)$ through (3)-(5). Assign to $R(x, z)$ a sequence of $n+1$ ($n := 2n_2$) 2-D polynomials

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

as follows.

Initiation. $E_0(x, z) = R(x, z) + z^n R(x, z^{-1})$

$$E_1(x, z) = \frac{R(x, z) - z^n R(x, z^{-1})}{z - 1}$$

Recursion. For $m = 0, 1, ..., n-2$: $zE_{m+2}(x, z) =$

$$\frac{e_{[m]\ 0}(x)(z+1)E_{m+1}(x, z) - e_{[m+1]\ 0}(x)E_m(x, z)}{\eta_{m-1}(x)}$$

where $\eta_m(x) = e_{[m]\ 0}(x)$ for $m \geq 1$, $\eta_m(x) = 1$ for $m < 1$.

Once again the matrices E_m are columnwise symmetric, i.e., $E_m J = E_m$, or $e_{[m]\ k}(x) = e_{[m]\ n-m-k}(x)$, $k = 0, ..., n-m$.

3.3 Stability Conditions for the E-Table

The next theorems state that stability conditions of form identical to those accompanying Algorithm 2 remain valid also for $\{E_m(x,z)\}$. Thus indeed the removed common factors are redundant and it is computationally more efficient to construct the reduced table.

Theorem 4. (Stability conditions for E-Table) *Assume $R(x,z)$ corresponds to $D(z_1,z_2)$ through (3)-(5) and that applying Algorithm 3 to it produced $\{E_m(x,z)\}_0^n$. The following conditions (i)-(iii) form a set of necessary and sufficient conditions for $D(z_1,z_2)$ to be stable:*
(i) $D(z_1,1) \neq 0 \quad \forall z_1 \in \bar{V}$
(ii) $D(1,z_2) \neq 0 \quad \forall z_2 \in \bar{V}$
(iii) $\epsilon_m(x) := E_m(x,1) \neq 0 \quad \forall x \in [-1,1] \quad m = 0,1,\ldots,n$

It is noted that the examination of these stability conditions that look like those for the previous "F-table" also gain from the simplification because now lower degree polynomials need to be examined. (We shall quantify these cost measures in [1]).

The next theorem establishes the even more remarkable result by which tighter stability condition, the counterpart of Theorem 3 for the F-table, hold also for the reduced "E-table". Again the result is stated currently without a proof [1].

Theorem 5. (Refined conditions for the E-table) *Assume $R(x,z)$ corresponds to $D(z_1,z_2)$ through (3)-(5) and that Algorithm 3 applied to it produced $\{E_m(x,z)\}$. The following conditions (i)-(iii) form a set of necessary and sufficient conditions for $D(z_1,z_2)$ to be stable:*
(i) $D(z_1,1) \neq 0 \quad \forall z_1 \in \bar{V}$
(ii) $D(1,z_2) \neq 0 \quad \forall z_2 \in \bar{V}$
(iii) $\epsilon_n(x) := E_n(x,1) = e_{[n]\ 0}(x) \neq 0 \, \forall x \in [-1,1]$

3.4 Concluding Remarks

A new stability testing method for 2-D polynomials has been presented. This conference version is an abridged form of a full paper [1]. The proofs omitted here will all become available in that paper. The procedure is readily programmed by an array oriented language like MATLAB by using the matricial/vector alternative presentation for 2-D/1-D polynomials and translation of polynomial multiplication/division to convolution/deconvolution. The full report [1] also contains more programming details, some routines, and evaluation of the order of complexity of the algorithm.

References

[1] Y. Bistritz, " An Immittance-Type Stability Test for 2-D Digital Filters" in preparation.

[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, H. Lev-Ari and T. Kailath, "Immittance domain Levinson algorithms" *IEEE Trans. Info. Th.* vol. 35, pp. 674-682, 1989.

[4] Y. Bistritz, H. Lev-Ari and T. Kailath, "Immittance type three-term Levinson and Schur recursions for quasi-Toeplitz complex Hermitian Matrices"' *SIAM J. on Matrix Anal. Appl.* vol. 12, pp. 497-520, 1991.

[5] Y. Bistritz, "A stability new test for linear discrete systems in table form" *IEEE Trans. Circ. Sys.* Vol. CAS-30, pp. 917-919,1983.

[6] N. K. Bose, "Implementation of a new stability test for two-dimensional filters", *IEEE Trans. on Acous. Speech, and Sig. Proc.*, Vol. ASSP-25,1977, pp. 117-120.

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

[8] E. I. Jury "Modified Stability Table for 2-D Digital Filter" *IEEE Trans. on Circ. Syst.* vol. CAS-35, 116-119, 1988.

[9] 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.

[10] B. M. Karan and M. C. Sarisvastava "A new test for 2-D filters" *IEEE Trans. Circ. Syst.*, vol. CAS-33, pp. 807-809, 1986.

[11] B. T. O'Connor and T. S. Huang, "Stability of general two-dimensional recursive filters", Ch. 4 in *Two-dimensional Digital Signal Processing I : Linear Filters*, (Ed. T. S. Huang) Springer-Verlag, Berlin 1981. Also in *IEEE Trans. on Acoust., Speech, and Sig. Proc.*, Vol. ASSP-26, pp. 550-560, December 1978.

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

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