CIRCUITS SYSTEMS SIGNAL PROCESSING
VOL. 22, NO. 3, 2003, PP. 255–276

REAL-POLYNOMIAL BASED IMMITTANCE-TYPE TABULAR STABILITY TEST FOR TWO-DIMENSIONAL DISCRETE SYSTEMS*

Yuval Bistritz[1]

Abstract. The paper proposes a new stability test for two-dimensional (2D) discrete systems. It is a tabular test; that is, it builds for the tested bivariate polynomial of degree (n_1, n_2) a sequence of n_2 (or n_1) bivariate polynomials or matrices (the "2D table") of increasing row and decreasing column sizes. It is an immittance-type test, which means that it uses a three-term recurrence relation to obtain a sequence of matrices with certain symmetry. It differs from some recent immittance tabular tests in that it is derived from the author's stability test for real polynomials instead of complex-coefficient polynomials. In comparison with related 2D stability tests developed before by Karan and Sarisvastava and by Premaratne, it simplifies the number of stability conditions and reduces the overall cost of computation from an exponential to a polynomial order of complexity.

Key words: Discrete-time systems, two-dimensional systems, stability testing, immittance algorithms.

1. Introduction

Stability assessment of two-dimensional (2D) discrete-time systems requires the testing of a bivariate polynomial

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

for the condition

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

where

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

---

* Received January 17, 2002; revised September 1, 2002.

[1] Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: bistritz@eng.tau.ac.il

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

A $D(z_1, z_2)$ that satisfies the condition (2) will be referred as two-dimensional stable. The exact relation of this definition to stability may be found in various texts on two-dimensional discrete systems, where references with particular focus on the stability issue include [21], [15], and [17]. Stability in the conventional case of one-dimensional (1D) discrete-time systems is well known to be related to the property that the characteristic polynomial has all its zeros in U. A polynomial with this property will also be called 1D stable. It is instrumental for developing methods to test (2) to realize that the condition of 1D stability can be expressed in a manner that looks like (2), viz.,

$$p(z) = \sum_{k=0}^{n} p_k z^k \neq 0 \qquad \forall z \in \bar{V} \tag{3}$$

Unfortunately, the converse is not possible. Namely, it is not possible to express the stability of 2D polynomials by confining the location of its zeros to U^2 (or to any other compact subspace of the double complex plane). The fact that numerical calculation of roots is not a viable alternative for testing 2D stability is already a strong incentive for developing algebraic stability tests for multidimensional systems. Algebraic tests for 2D systems also provide useful design tools, much like their role in 1D systems, e.g., for deriving constraint ranges on symbolic parameters. Finally, efficient and elegant algorithms for testing the stability of multidimensional systems might have an impact on related multidimensional signal processing algorithms similar to the interesting relations that are known between algorithms for testing the stability of one-dimensional systems and some important signal processing algorithms.

Although the problem has been studied by now for more than three decades, until quite recently the available algebraic solutions for testing the condition (2) used to have an exponential order of complexity governed by the term $n^2 2^{2n}$ for $n = n_1 = n_2$. (We shall usually use the assumption $n = n_1 = n_2$ to simplify reference to the complexity of 2D stability testing techniques.) The method to be presented in this paper is elaborated from an earlier conference presentation [5]. It was the first paper that reported a tabular test that steps down from exponential to polynomial complexity of order n^6. The tabular test in [16], which is based on the modified Jury test (MJT) for 1D stability of complex polynomials in [18], also has order n^6 complexity, as was shown more recently in [9]. Other 2D stability tests that attain order n^6 were reported in [6] and [7]. They differ in various details from the test presented here because they stem from the form of the Bistritz test (BT) for complex polynomials and not for real polynomials as is used here.

This paper focuses on developing a 2D stability test from the BT for real-coefficient polynomials in [3]. This task was also considered before by Karan and Sarisvastava [19], [20] and by Premaratne [22]. The paper will improve the tests presented in these references by obtaining a simpler and smaller set of stability conditions and by significantly reducing the overall cost of computation.

The derivation of the 2D stability test in this paper will be carried out in two parts. The paper starts with some preliminaries in Section 2. Then Section 3 introduces a first form of the test, its properties, including the use of polynomial recursion, and a single positivity condition. The capacity of this first form covers or exceeds the results attained in [19], [20], and [22]. It is also shown in Section 3 that this form has an exponential order of computational complexity. Next, in Section 4, the first form is simplified into a test of n^6 order of complexity by exhibiting the existence of certain redundant common factors and eliminating them. The resulting simplified test is the main result of this paper. Even though the complexity has been reduced, the main form retains the simple first form of the stability conditions. A more thorough count of computation for the new test is held in Section 5, and its outcome is compared with those of other efficient tabular tests. Accordingly, its efficiency compares well with the best available scattering-type tabular tests in [16], [9], but it requests twice the cost of the immittance tabular tests in [6] and [7] that are based on the complex BT. Section 5 also summarizes the procedure and illustrates it by a numerical example. The paper ends with some comments. The most important of them draws attention to the fact that this test forms the basis of an efficient 2D stability test of $O(n^4)$ that, unlike all other tests of comparable order of complexity, involves only real arithmetics and real 1D stability tests.

2. Preliminaries

This section first introduces the notational conventions and then provides some required auxiliary results (part of which are derived in the Appendix).

2.1. Notation

The paper will consider only real (i.e., real-coefficient) polynomials. The polynomial variables will be denoted by s and x (usually implying affiliation with values on T and in the real interval $[-1, 1]$, respectively) or else by z. A polynomial like (1) will be said to have degree (n_1, n_2), and it may also be written as $D(z_1, z_2) = \mathbf{z_1}^t D\mathbf{z_2}$, where $\mathbf{z} := [1, z, \ldots, z^i, \ldots]^t$ is a vector of length depending on context and D is also used to denote its coefficient matrix. Similarly, for a univariate polynomial $p(z)$, p also denotes its coefficient vector and it may also be written as $p(z) = \mathbf{z}^t p$. Reversion is defined for a column vector by $p^\sharp := Jp$, where J denotes the reversion matrix (a square matrix with 1's on its main antidiagonal and zeros elsewhere) and reversion for a polynomial by $p^\sharp(z) = \mathbf{z}^t p^\sharp$. A vector and a polynomial are called symmetric if $p = p^\sharp$ and $p(z) = p^\sharp(z)$. Reversion of only columns in a matrix will be denoted by $D^\flat = DJ$. A matrix F and its corresponding polynomial $F(x, z)$ will be said to posses column symmetry if $F^\flat = F$. Indices are attached to matrices to denote their position in a sequence of polynomials (that constitutes the "2D table"), e.g., $F_m(x, z) = \mathbf{x}^t F_m \mathbf{z}$, $m = 0, \ldots, n$. Reference to columns of an indexed matrix carries its identifying index in brackets, $F_m = [f_{[m]\,0}, f_{[m]\,1}, \ldots]$. Indices will also be used for a sequence of univariate polynomials (when they form the 1D table), $f_m(z) = \mathbf{z}^t f_m = \sum_{i=0}^{n-m} f_{m,i} z^i$, $m = 0, \ldots, n$. Finally, for a polynomial $p(s) = [1, s, \ldots, s^n]p$, we define $p(\tilde{s}) = s^{-n/2} p(s)$, as its *balanced polynomial*. The latter may also be expressed as $\tilde{\mathbf{s}}^t p$, where $\tilde{\mathbf{s}} = [s^{-n/2}, s^{-(n-1)/2}, \ldots, s^{(n-1)/2}, s^{n/2}]^t$.

2.2. Auxiliary results

The most commonly used starting point for developing 2D stability tests has been a simplification of the stability condition (2) due to Huang and Strintzis (see [21] and [17] for details and references).

Lemma 1. *$D(z_1, z_2)$ is stable (i.e.,* (1) *obeys* (2)*) if and only if*

$$D(z, a) \neq 0 \quad \text{for all} \quad z \in \bar{V} \text{ and some } a \in \bar{V} \tag{4}$$

and

$$D(s, z) \neq 0 \quad \text{for all} \quad (s, z) \in T \times \bar{V}\ . \tag{5}$$

This lemma admits the testing of (2) by any algebraic method that can test the condition (3) for a complex (i.e., complex coefficient) polynomial. The general idea is to regard $D(s, z)$ as a univariate polynomial in the variable z with complex coefficients dependent on $s \in T$, viz.,

$$D(s, z) = \sum_{i=0}^{n_2} d_i(s) z^i, \tag{6}$$

where, in the above, the coefficient matrix D is presented by its columns $D = [d_1, d_2, \ldots, d_{n_2}]$. In order to develop 2D stability tests from 1D stability conditions available for polynomials with *real* coefficients, Bose proposed in [14] (see also [15]) to multiply $D(s, z)$ by its complex conjugated coefficients polynomial for $s \in T$, $D(s^{-1}, z)$,

$$\tilde{\mathbf{s}}^t Q \mathbf{z} := D(s^{-1}, z) D(s, z) = \sum_{k=0}^{2n_2} q_k(\tilde{s}) z^k. \tag{7}$$

Afterwards, it is possible to use the transformation

$$x = \frac{s^{-1} + s}{2}, \tag{8}$$

to convert the degree $(2n_1, 2n_2)$ polynomial $Q(s, z)$ into

$$H(x, z) = Q(\tilde{s}, z)\Big|_{x = \frac{1}{2}(s^{-1}+s)}, \tag{9}$$

which has degree $(n_1, 2n_2)$. Because the transformation (8) maps the unit circle T to the real interval $[-1, 1]$, Lemma 1 is "mapped" into the following lemma.

Lemma 2. *$D(z_1, z_2)$ is stable if and only if*

$$D(z, a) \neq 0 \quad \text{for all} \quad z \in \bar{V} \quad \text{and some } a \in \bar{V} \tag{10}$$

and

$$H(x, z) \neq 0 \quad \forall x \in [-1, 1] \quad \text{and} \quad \forall z \in \bar{V}. \tag{11}$$

Lemma 2 admits the development of 2D stability tests using real 1D stability tests. In the following we shall use it with $a = 1$, because $z = 1$ plays further roles in the standard form of the immittance formulation setting.

The conversion of $D(z_1, z_2)$ into $H(x, z)$ can be treated as an algebraic transformation between the coefficients. The derivation is given in the Appendix, and its outcome is described in Section 5 in the summary of the proposed test.

3. Intermediate form of the test

In this section, first a division-free form for the 1D stability test [3] is presented. Next, Lemma 2 is used to obtain a corresponding 2D tabular test. Then, the initial and obvious necessary and sufficient conditions for stability posed on it are refined to contain only a single positivity test of a univariate polynomial over $[-1, 1]$. Finally, the computation cost of the preliminary test form is shown to be of exponential order of complexity.

3.1. Division-free real 1D stability test

An obvious start for developing a 2D stability test from the real 1D polynomial stability test [3] would be to use it to test the condition (11) in Lemma 2 regarding $H(x, z)$ as a univariate polynomial in z with coefficients dependent on x. However, this approach leads to a polynomial in z whose coefficients are a rational function in x as in [19]. This is an unnecessary complication that we shall avoid from the outset by first replacing the stability test in [3] by a division-free version as follows.

Algorithm 1. Division-free 1D table. Assume a real polynomial $p(z) = \sum_{k=0}^{n} p_k z^k$ with $p(1) > 0$ and assign to it a sequence $\{f_m(z) = \sum_{k=0}^{n-m} f_{m,k} z^k,\ m = n, \ldots, 0\}$ of symmetric polynomials ($f_{m,n-m-i} = f_{m,i}$, $i = 0, \ldots, n - m$) as follows.

(i) **Initiation**

$$f_0(z) = p(z) + p^\sharp(z), \quad f_1(z) = \frac{p(z) - p^\sharp(z)}{(z - 1)}. \tag{12}$$

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

$$z f_{m+2}(z) = f_{m,0}(z + 1) f_{m+1}(z) - f_{m+1,0} f_m(z). \tag{13}$$

Theorem 3. (Stability criterion for Algorithm 1). *Assume* $p(z)$ *such that* $p(1) > 0$ *and that Algorithm 1 produced for it the sequence* $\{f_m(z)\}_0^n$.
(a) $p(z)$ *is stable if and only if*

$$f_m(1) > 0 \qquad m = 0, 1, \ldots, n. \tag{14}$$

(b) If $p(z)$ *is stable, then the following conditions:*

$$f_{m,0} > 0 \qquad m = 0, 1, \ldots, n \tag{15}$$

hold as well.

This form of the stability test can be proved from the stability conditions in [3] and the observation that the sequence of polynomials here and there differ by positive scaling factors. The details will be made available elsewhere [12].

3.2. Corresponding 2D tabular test

The next algorithm implements the application of Algorithm 1 to $H(x, z)$ of (9) regarding it as a univariate polynomial in the variable z of degree $n := 2n_2$ with coefficients that are univariate polynomials in x, viz., $H(x, z) = \sum_0^n h_i(x)z^i$.

Algorithm 2. The "F-table" (a preliminary table form). Convert $D(z_1, z_2)$ into $H(x, z)$ and assign to the latter a sequence

$$F_m(x, z) = \sum_{k=0}^{n-m} f_{[m]\,k}(x)z^k, \quad m = 0, \ldots, n := 2n_2, \tag{16}$$

of column-symmetric bivariate polynomials, $F_m = F_m^\flat$ (i.e., $f_{[m]\,k}(x) = f_{[m]\,n-m-k}(x)$, $k = 0, \ldots, n - m$), constructed as follows.

(i) **Initiation**

$$F_0(x, z) = H(x, z) + H^\flat(x, z), \quad F_1(x, z) = \frac{H(x, z) - H^\flat(x, z)}{z - 1}. \tag{17}$$

(ii) **Recursion.** For $m = 0, 1, \ldots, n - 2$, do:

$$zF_{m+2}(x, z) = f_{[m]\,0}(x)(z + 1)F_{m+1}(x, z) - f_{[m+1]\,0}(x)F_m(x, z). \tag{18}$$

Remark 1. The requirement $p(1) > 0$ in Algorithm 1 translates into $H(x, 1) = Q(\tilde{s}, 1) = D(s^{-1}, 1)D(s, 1) > 0$. It clearly holds if $D(z, 1)$ is 1D stable. Stability of $D(z, 1)$, a clear necessary condition for 2D stability, will be part of the test, and its testing should of course precede the starting of Algorithm 2.

Theorem 4. (Stability conditions for the F-table). *Let* $\{F_m(x, z)\}_0^n$ *be the sequence that Algorithm 2 assigns to* $D(z_1, z_2)$.

(a) The following conditions (i), (ii), and (iii), or (iii′) form a set of necessary and sufficient conditions for $D(z_1, z_2)$ to be stable:

$$(i)\quad D(z_1, 1) \neq 0 \quad \forall z_1 \in \bar{V} \tag{19}$$

$$(ii)\quad D(1, z_2) \neq 0 \quad \forall z_2 \in \bar{V} \tag{20}$$

$$(iii)\quad \varphi_m(x) := F_m(x, 1) \neq 0 \quad \forall x \in [-1, 1] \text{ and } m = 0, 1, \ldots, n \tag{21}$$

$$(iii')\quad \varphi_m(x) > 0 \quad \forall x \in [-1, 1] \quad \text{and} \quad m = 0, 1, \ldots, n. \tag{22}$$

(b) If $D(z_1, z_2)$ is stable, then the next conditions

$$f_{[m]\,0}(x) > 0 \qquad \forall x \in [-1, 1] \qquad m = 0, 1, \ldots, n \tag{23}$$

also hold.

Proof. If $D(z_1, z_2)$ is stable, then (i) and (ii) hold as "special cases." Regard $D(s, z)$ as a univariate polynomial in z that is stable for all $s \in T$, to obtain that $Q(\tilde{s}, z) = D(s, z)D(s^{-1}, z)$ is also 1D stable $\forall s \in T$. Thus, $H(x, z)$ is 1D stable for all $x \in [-1, 1]$. The necessity of the conditions (iii′) follows now from the necessity of (14) in Theorem 3.

Conversely, assume that the three conditions (i), (ii), (iii) hold. Condition (ii) implies that $Q(1, z) = D(1, z)^2$ is stable. Therefore, $H(x, z)$ is 1D stable for $x = 1$. Then, by Theorem 3, all $\varphi_m(1) > 0$. The latter together with condition (iii) imply that (iii′) holds. Because (14) in Theorem 3 is sufficient there for 1D stability, it follows that $H(x, z)$ is 1D stable for all points $x \in [-1, 1]$; that is, (11) holds. Adding to the batch the condition (10) of Lemma 2, in the form of condition (i), implies that $D(z_1, z_2)$ is 2D stable.

It remains to prove (23). Set $z = 1$ in (18) to obtain

$$2\varphi_{m+1}(x) f_{[m]\,0}(x) = \varphi_{m+2}(x) + f_{[m+1]\,0}(x)\varphi_m(x). \tag{24}$$

If $D(z_1, z_2)$ is 2D stable, then (22) holds. Using it in (24) shows that $f_{[m+1]\,0}(x) > 0$ implies $f_{[m]\,0}(x) > 0$. Starting with $f_{[n]\,0}(x) = \varphi_n(x) > 0$ and proceeding backward proves (23). □

Remark 2. It follows from the proof of Theorem 4 that necessary and sufficient conditions are given by (i), (ii), and (iii) or equivalently by (i) and (iii′). This makes (i) and (iii′) a seemingly more compact set of conditions. However, there is no algebraic method that can examine (iii′) any better than testing (iii). Also, there is no reason to drop (ii) because testing it before starting Algorithm 2 (for a relatively negligible order n^2 cost) will save the computationally more demanding Algorithm 2 whenever (ii) is found not to hold.

The development so far has attained a level comparable to that in [19], [20]. Some notable differences (beyond those caused by using there a slightly different version of the test) are that here we work with polynomials only and not rational functions, and that the size of the set of necessary and sufficient conditions for stability in Theorem 4 is already less than half of the number of conditions in [19], [20].

3.3. Refined stability conditions

Next, we proceed to show that 2D stability can be concluded after verifying the no zeros on $[-1, 1]$ (or positivity) condition for only the last of the polynomials of the 2D table. This refinement is similar in spirit to the simplification Siljak derived for determinantal type 2D tests [24]. However, it will be obtained here from inherent properties of the recursion without reference to determinants or to any other extraneous argument.

Theorem 5. (Refined stability conditions for the F-table). *Let* $\{F_m(x, z)\}_0^n$ *be the sequence that Algorithm 2 assigns to* $D(z_1, z_2)$*. The following conditions (i), (ii), and (iii) or (iii′) are necessary and sufficient for* $D(z_1, z_2)$ *to be stable:*

$$(i)\quad D(z_1, 1) \neq 0 \qquad \forall z_1 \in \bar{V} \tag{25}$$

$$(ii)\quad D(1, z_2) \neq 0 \qquad \forall z_2 \in \bar{V} \tag{26}$$

$$(iii)\quad \varphi_n(x) \neq 0 \qquad \forall x \in [-1, 1] \tag{27}$$

$$(iii')\quad \varphi_n(x) > 0 \qquad \forall x \in [-1, 1], \tag{28}$$

where $\varphi_n(x) = F_n(x, 1)$.

Proof. Necessity is obvious because the conditions here are a subset of the conditions in Theorem 4. Sufficiency will follow from Theorem 4 after we show that conditions (i)–(iii) imply that $\varphi_m(x) \neq 0\ \forall x \in [-1, 1]$ also for $m < n$. Assume then that conditions (i)–(iii) hold, but that nevertheless there exists a $\varphi_m(x)$ that vanishes for some values in the interval $[-1, 1]$. Note that (ii) guarantees via Theorem 3 that $\varphi_m(1) > 0$ and $f_{[m]\,0}(1) > 0$ for all $m = 0, 1, \ldots, n$. Let k_1 be the least m such that either ("case 1":) $\varphi_{k_1}(x) = 0$ or ("case 2":) $f_{[k_1]\,0}(x) = 0$ vanish for some $x \in [-1, 1]$ and let the maximal such root in $[-1, 1]$ be $x_1 < 1$. If $\varphi_{k_1}(x_1) = 0$ ("case 1"), then use (24) with $k_1 = m + 1$ to obtain

$$\varphi_{k_1+1}(x_1) = -f_{[k_1]\,0}(x_1)\varphi_{k_1-1}(x_1).$$

By the assumptions on k_1, $\varphi_{k_1-1}(x_1) > 0$ and $f_{[k_1]\,0}(x_1) \geq 0$; thus, it follows that $\varphi_{k_1+1}(x_1) \leq 0$. If so, then $\varphi_{k_1+1}(x)$ must have a zero in $[x_1, 1)$. Else, if $f_{[k_1]\,0}(x_1) = 0$ ("case 2"), then use (24) with $k_1 = m$ to obtain

$$\varphi_{k_1+2}(x_1) = -f_{[k_1+1]\,0}(x_1)\varphi_{k_1}(x_1).$$

Then, because $\varphi_{k_1}(x_1) > 0$, either $\varphi_{k_1+2}(x)$ or $f_{[k_1+1]\,0}(x)$ is nonpositive at x_1 and therefore has a zero in $[x_1, 1)$. Thus, the assumption on k_1 implies (in both cases) that there exists $k_2 > k_1$ such that either $\varphi_{k_2}(x) = 0$ or $f_{[k_2]\,0}(x) = 0$ for some x in $[x_1, 1)$. Let x_2 denote the maximal of these roots, $x_1 \leq x_2 < 1$. The argument can now be repeated with k_2 and x_2 replacing k_1 and x_1, and it leads to the fact that there exists a next $k_3 > k_2$ and $x_3 \geq x_2$ that is the maximal zero that either $\varphi_{k_3}(x)$ or $f_{[k_3]\,0}(x)$ has in the subinterval $[x_2, 1)$. After enough such repetitions, say $k_\ell (k_\ell \leq n-k_1)$, the conclusion will read: "either $\varphi_n(x)$ or $f_{[n]\,0}(x)$ must vanish in $[x_{k_\ell -1}, 1]$ for some x_{k_ℓ} where $-1 \leq x_1 \leq x_{k_\ell -1} \leq x_{k_\ell} < 1$." This is a contradiction to assumption (iii) (recall that $f_{[n]\,0}(x) = \varphi_n(x) = F_n(x, z)$). The contradiction proves that conditions (i)–(iii) do imply that $\varphi_m(x) \neq 0 \; \forall x \in [-1, 1]$ also for all $m < n$. □

The progress of the paper until this point is comparable to the refinements that were obtained by Premaratne in [22]. He showed there that the test proposed in [19], [20] can be completed while working only with polynomials and he too obtained the single positivity condition. He derived the latter condition by combining relations that he obtained between the BT and the Schur-Cohn minors in [23] with Siljak's simplification [24]. Note that here the single positivity test arises from inherent properties of the recursion. Some minor differences are also noticed if one compares Theorem 5 with its closest parallel, Theorem 14 in [22].

3.4. *Cost of computation*

The computational complexity for the tabular test presented so far can be obtained as follows. Let $\ell_f(m)$ denote the x variable degree of $F_m(x, z)$. It begins with $\ell_f(0) = n_1$, $\ell_f(1) = n_1$, then obeys a Fibonacci difference equation $\ell_f(m) = \ell_f(m-1) + \ell_f(m-2)$ whose solution, subject to the given initial conditions, is $\ell_f(m) = n_1(\lambda_1^{m+1} - \lambda_2^{m+1})/(\lambda_1 - \lambda_2)$, where $\lambda_1 = 1/2 + \sqrt{5}/2$ and $\lambda_2 = 1/2 - \sqrt{5}/2$. The creation of $F_m(x, z)$ requires $(n-m+1)$ times the multiplication of a polynomial of degree $\ell_f(m-1)$ by one of degree $\ell_f(m-2)$ for all $m = 2, \ldots, n := 2n_2$ (with attention to the column symmetry already accounted). The multiplication of two polynomials of degrees k_1 and k_2 requires $(k_1+1)(k_2+1)$ operations (real "flops," where a flop is one addition plus one multiplication). Following this prescription, the exact count of flops can be worked out. It suffices to realize that the cost is $O(n_1^2 \lambda_1^{4n_2})$ (where the "big Oh" is used to cite only the term in the full expression that becomes the most dominant for large n_1 and n_2). The cost of testing the one positivity test has a similar order of complexity. Testing one (or even all) positivity test(s) takes only a small fraction of the cost of computation required for the construction of the table.

The cause for the current exponential complexity is analyzed in the next section. The analysis leads to an improved form of the tabular test that requires a significantly lower amount of computation.

4. Final form of the test

This section first reveals that Algorithm 2 creates a sequence $F_m(x, z)$ that tends to accumulate redundant factors in the x variable. Then, a revised recursion that produces a sequence $\{R_m(x,z)\}$ of bivariate polynomials that is free from these common factors is presented. The revised recursion is shown to obey stability conditions of equally compact and simple form, and the overall computation cost of the new stability test is shown to be significantly reduced.

4.1. Redundant common factors

The next lemma shows that the recursion (18) produces common x polynomial factors. The characterization it provides on the pattern of their creation and accumulation will be useful afterwards for their elimination.

Lemma 6. *Consider a sequence* $\{F_m(x,z)\}_0^n$ *that is produced by Algorithm 2.*

(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) *2 is a factor of* $F_2(x,z)$.
(c) *For any subset of consecutive polynomials in this sequence:* $F_m(x,z)$, $F_{m+1}(x,z)$, $F_{m+2}(x,z)$, $F_{m+3}(x,z)$, $F_{m+4}(x,z)$, $0 \geq m$ *and* $m+4 \leq n$, $f_{[m+1]\,0}(x)$ *is a factor of* $F_{m+4}(x,z)$.

The proof of this lemma will not be given here. It follows from related results that will become available in [12]. The lemma makes the significant observation that each polynomial $F_{3+k}(x,z)$ in Algorithm 2, $k \geq 1$, contains multiplicities of all of the following factors $f_{[i]\,0}(x), \quad i = 1,\ldots,k$ and that these factors build up rapidly. Property (b) is somewhat different; it does not affect the growth of the degree in x of the polynomials, but it still affects their scaling. Its inclusion therefore does not reduce the growth of polynomial degrees in x but avoids an otherwise rapid growing (as powers of 2) of the scalar scaling factors of the polynomials (see [12]).

4.2. Reduced size table

One obvious way to handle these factors could be removing them at the end, after completing the "F-table." This approach would not reduce but actually increase computational complexity (even though it reduces the degree of the lastly tested polynomial). Fortunately, Lemma 6 also indicates how these factors can be removed recursively as soon as they occur. The next algorithm provides a recursion that produces a sequence of bivariate polynomials $R_m(x,z)$ such that, for each m, $R_m(x,z)$ corresponds to $F_m(x,z)$, after dividing out from it all the factors $f_{[i]\,0}(x), i \leq m$. Lemma 6 guarantees that $f_{[i]\,0}(x), i \leq m$, are common to all $f_{[m]\,k}(x)$, $k = 0, 1, n-m$, so that in the following algorithm all the produced $R_m(x,z)$ are polynomial.

Algorithm 3. The 2D real table. Convert $D(z_1, z_2)$ into $H(x, z)$ and assign to it a sequence

$$R_m(x, z) = \sum_{k=0}^{n-m} r_{[m]\,k}(x) z^k \quad m = 0, \ldots, n := 2n_2, \tag{29}$$

of column-symmetric bivariate polynomials, $R_m = R_m^\flat$ (i.e., $r_{[m]\,k}(x) = r_{[m]\,n-m-k}(x)$, $k = 0, \ldots, n - m$), that is constructed as follows.

(i) **Initiation**

$$R_0(x, z) = H(x, z) + H^\flat(x, z), \quad R_1(x, z) = \frac{H(x, z) - H^\flat(x, z)}{z - 1}. \tag{30}$$

(ii) **Recursion.** For $m = 0, 1, \ldots, n - 2$:

$$zR_{m+2}(x, z) = \frac{r_{[m]\,0}(x)(z + 1)R_{m+1}(x, z) - r_{[m+1]\,0}(x)R_m(x, z)}{\eta_m(x)}, \tag{31}$$

where $\eta_0(x) = 2$, $\eta_1(x) = 1$, and

$$\eta_m(x) = r_{[m-1]\,0}(x) \quad \text{for} \quad m \geq 2. \tag{32}$$

Remark 3. The key to improving the efficiency of the preliminary table form is the exposition and elimination of the common factors that the preliminary recursion form tends to accumulate. A similar observation also led to the lowering of the cost of computation from exponential to polynomial complexity for the tabular tests in [16], [9], [6], [7]. However, the details of the underlying mechanism that enables such complexity reductions involve tricky differences. It caused the author in [22], after finding that a mechanism that works for the scattering formulation (the modified Jury test, MJT) does not work for the immittance formulation (the BT), to declare that "we have exposed a drawback in using the BT (...) — a rapid growth of the degree of each polynomial entry [that] can not be avoided (...) in contrast to the MJT." To make things even more baffling, the common factors take a different shape, and their elimination involves a different mechanism, not only when the immittance and the scattering formulations are compared, but also when the real and complex forms of the immittance formulation are compared. The common factor in the complex immittance tests in [7] and [6] look different from their form here. Also, here the factor $r_{[m]\,0}$ is removed when (and not earlier) $R_{m+3}(z)$ is created. In the case of the complex recursion, a factor related to the mth bivariate polynomial can be removed when the $(m + 2)$-th bivariate polynomial is created, i.e., one step earlier.

4.3. Stability conditions for the reduced size table

The derivation of stability conditions for the reduced table will use the conditions obtained for the "F-table." The two sequences, $\{F_m(x, z)\}$ and $\{R_m(x, z)\}$, differ by a sequence of multiplying factors, say $\alpha_m(x)$,

$$F_m(x,z) = \alpha_m(x) R_m(x,z). \tag{33}$$

Assume first that $F_m(x,z)$'s are known. Substitute (33) into (18) and compare the outcome with (31) to obtain that $\alpha_0(x) = 1$, $\alpha_1(x) = 1$, $\alpha_2(x) = 2$, and afterwards,

$$\alpha_{m+2}(x) = \frac{\alpha_{m+1}(x)\alpha_m(x)}{\alpha_{m-1}(x)} f_{[m-1]\,0}(x), \quad m \geq 1. \tag{34}$$

Conversely, when $R_m(x,z)$'s are known and the $F_m(x,z)$ are to be obtained, set $f_{[m]\,0}(x) = \alpha_m(x) r_{[m]\,0}(x)$ into (34), to see that the $\alpha_m(x)$ can be constructed from the $r_{[m]\,0}(x)$'s, starting with $\alpha_0(x) = 1$, $\alpha_1(x) = 1$, $\alpha_2(x) = 2$, and following with

$$\alpha_{m+2}(x) = \alpha_m(x)\alpha_{m+1}(x) r_{[m-1]\,0}(x), \quad m \geq 1. \tag{35}$$

To proceed, we single out the polynomials that present the values at $z = 1$ of the 2D sequence $\{R_m(x,z)\}$ and denote them by

$$\rho_m(x) := R_m(x,1) = \sum_{k=0}^{n-m} r_{[m]\,k}(x), \quad m = 0, 1, \ldots, n. \tag{36}$$

The relations (33)–(35) between the polynomials $R_m(x,z)$ and $F_m(x,z)$ in the two tables, in combination with Theorem 4, make it apparent that a possible set of necessary and sufficient conditions for stability for the reduced table consists of the conditions (i) and (ii) in Theorem 4 plus replacing (iii) there by the following conditions:

$$\rho_m(x) > 0, \quad r_{[m]\,0}(x) > 0 \quad \forall x \in [-1,1], \quad m = 0, 1, \ldots, n. \tag{37}$$

This set of conditions has twice the size of the initial stability condition for $\{F_m(x,z)\}$ in Theorem 4. One might wonder whether it is possible to drop from this set the $r_{[m]\,0}(x) > 0$ conditions and by doing so to draw them close in form and number to the conditions in Theorem 4. To pursue this possibility, we set $z = 1$ in (31) to obtain the following relations between the set of the $r_{[m]\,0}(x)$'s and the set of $\rho_m(x)$'s:

$$\rho_{m+2}(x) = \frac{2r_{[m]\,0}(x)\rho_{m+1}(x) - r_{[m+1]\,0}(x)\rho_m(x)}{\eta_m(x)} \tag{38}$$

with $\eta_m(x) = r_{[m-1]\,0}(x)$ for $m \geq 2$ as defined in (32). What may hinder the disposal of the condition $r_{[m]\,0}(x) > 0$ is the concern that simultaneously negative values at the numerator and denominator in (38) for values of $x \in [-1,1]$ may produce a positive $\epsilon(x)$, thus hiding the detection of instability that such negative values would otherwise disclose. The next theorem resolves this concern.

Theorem 7. (Preliminary stability conditions for Algorithm 3). *Let* $\{R_m(x,z)\}$ *be the sequence that Algorithm 3 produces for* $D(z_1, z_2)$.

(a) The following conditions (i), (ii), and (iii) or (iii′) are necessary and sufficient conditions for $D(z_1, z_2)$ to be stable:

$$(i)\ \ D(z_1, 1) \neq 0 \qquad \forall z_1 \in \bar{V} \tag{39}$$

$$(ii)\ \ D(1, z_2) \neq 0 \qquad \forall z_2 \in \bar{V} \tag{40}$$

$$(iii)\ \ \rho_m(x) \neq 0 \quad \forall x \in [-1, 1] \quad m = 0, 1, \ldots, n \tag{41}$$

$$(iii')\ \ \rho_m(x) > 0 \quad \forall x \in [-1, 1] \quad m = 0, 1, \ldots, n. \tag{42}$$

(b) If $D(z_1, z_2)$ is stable, then the conditions

$$r_{[m]\,0}(x) > 0 \quad \forall x \in [-1, 1], \qquad m = 0, 1, \ldots, n \tag{43}$$

also hold.

Proof. The fact that the 2D stability implies the conditions in (i), (ii), (iii), (iii′) and (b) was already concluded (see (37)).

To prove the converse, we first show that the conditions (i)–(iii) imply (43). Assume the conditions (i)–(iii), then $R(1, z) \neq 0\ \forall z \in \bar{V}$ because $R(1, z) = D(1, z)^2$ is stable. Thus the "F-table" produces $\varphi_m(x)$ and $f_{[m]\,0}(x)$ that are positive at $x = 1$ for all m. Use (35) and (33) to conclude that $\rho_m(x)$ and $r_{[m]\,0}(x)$ are then positive at $x = 1$. To prove (43), assume to the contrary, that not all $r_{[m]\,0}(x) > 0\ \forall x \in [-1, 1]$. Then let $r_{[k]\,0}(x)$ be the first $r_{[m]\,0}$ (i.e., k is the lowest index m) such that $r_{[k]\,0}(x)$ becomes negative on $[-1, 1]$ and let x_k be its rightmost zero on this interval. Then use (38) to obtain

$$\rho_{k+2}(x_k) = -r_{[k+1]\,0}(x_k)\rho_k(x_k)/\eta_k(x_k).$$

This implies that $r_{[k+1]\,0}(x_k) < 0$. Let x_{k+1} be the rightmost zero of $r_{[k+1]\,0}(x)$ in $[-1, 1]$, $x_k < x_{k+1}$. It follows then that

$$\rho_{k+3}(x_{k+1}) = -r_{[k+2]\,0}(x_{k+1})\rho_{k+1}(x_{k+1})/\eta_{k+1}(x_{k+1}).$$

This implies that $r_{[k+2]\,0}(x_{k+1}) < 0$. This reasoning can be repeated until one obtains that there exists an $x_{n-2} \in [-1, 1]$ for which $r_{[n-2]\,0}(x_{n-2}) = 0$, with $x_{k+1} < x_{k+2} < \ldots < x_{n-3} < x_{n-2} < 1$, for which $r_{n-3}(x_{n-2}) > 0$ and therefore

$$\begin{aligned}\rho_n(x_{n-2}) &= -r_{[n-1]\,0}(x_{n-2})\rho_{n-2}(x_{n-2})/\eta_{n-2}(x_{n-2}) \\ &= -0.5\rho_{n-1}(x_{n-2})\rho_{n-2}(x_{n-2})/r_{[n-3]\,0}(x_{n-2}) < 0.\end{aligned}$$

The last conclusion contradicts the assumptions, because if (i)–(iii) hold, then (iii′) holds. Thus the hypothesis that *not* all $r_{[m]\,0}(x)$ are positive $\forall x \in [-1, 1]$ is false. This proves that (i), (ii), and (iii) imply (43).

Assume that (i), (ii), and (iii) are true. Then (43) is true. Therefore, all $\alpha_m(x) > 0\ \forall x \in [-1, 1]$ via (35). Then all $\varphi_m(x) > 0\ \forall x \in [-1, 1]$ (33) and stability of $D(z_1, z_2)$ follows from Theorem 4 (or say from Theorem 5 because in particular one has that $\varphi_n(x) > 0\ \forall x \in [-1, 1]$). □

The next theorem sharpens the stability conditions that can be associated with the sequence $\{R_m(x, z)\}$ to conditions that look like the refined condition for $\{F_m(x, z)\}$ in Theorem 5.

Theorem 8. (Concise stability conditions for Algorithm 3). *Let* $\{R_m(x,z)\}$ *be the sequence that Algorithm 3 produced for* $D(z_1,z_2)$. *The following conditions (i), (ii), and (iii) or (iii′) form a set of necessary and sufficient conditions for* $D(z_1,z_2)$ *to be stable:*

$$(i)\ \ D(z_1,1)\neq 0 \qquad \forall z_1\in\bar{V} \tag{44}$$

$$(ii)\ \ D(1,z_2)\neq 0 \qquad \forall z_2\in\bar{V} \tag{45}$$

$$(iii)\ \ \rho(x)\neq 0 \qquad \forall x\in[-1,1] \tag{46}$$

$$(iii')\ \ \rho(x)>0 \qquad \forall x\in[-1,1], \tag{47}$$

where $\rho(x) := R_n(x,z)$ *is the last polynomial in the sequence produced by Algorithm 3.*

Proof. The stated conditions are necessary for 2D stability because they form a subset of the conditions in Theorem 7.

Sufficiency is proved if one shows that the stated conditions imply the larger set of sufficient conditions in Theorem 7. More specifically, it has to be shown that $\rho_n(x)\neq 0\ \forall x\in[-1,1]$ implies $\rho_m(x)\neq 0\ \forall x\in[-1,1]$ also for $m<n$. A rigorous full proof involves an argument that, similar to what was employed to prove Theorem 5 from Theorem 4, examines simultaneously the possible propagation of zeros of either $r_{[k]\,0}(x)$ and $\rho_m(x)$ down the recursion. For brevity, we shall only explain why, in spite of the differences between the two recursions, the argument used to prove Theorem 5 still works.

First notice that the proof by contradiction of the sufficiency part for Theorem 5 relies on examining whether zeros of $f_{[i]\,0}(x)$ or $\varphi_i(x)$ in shrinking subintervals of $[-1,1]$ may be transmitted downward in the recursion. It was shown there that the transmission of possible zeros of $f_{[i]\,0}(x)$ or $\varphi_i(x)$ is completed in *at most two steps*. Next recall that the $R_m(x,z)$ polynomial differs from its corresponding $F_m(x,z)$ by elimination of factors $f_{[i]\,0}(x)$ that belong to $F_i(x,z)$ of *three or more steps* upward $i\leq m-3$; see (33), (34). Thus, if a polynomial $f_{[i]\,0}(x)$ has a zero, say x_i in the interval $[-1,1]$, it would survive in the recursion (31) long enough to "infect" a term further down in the recursion to have a zero, say $x_i\leq x_{i+1}<1$ before it has been eliminated. In other words, a proof by contradiction, similar to the one detailed to prove sufficiency of the conditions in Theorem 5, can also be used here and would show that no zeros in $[-1,1]$ for $\rho_n(x)$ implies $\rho_m(x)\neq 0$ $\forall x\in[-1,1]$ also for $m<n$. □

Algorithm 3 and Theorem 8 constitute the 2D tabular stability test that this paper proposes. Of course, the conditions in Theorem 7 and those in (37) that are not part of the final concise set of conditions of Theorem 8 are further necessary conditions for 2D stability.

5. An outline, cost analysis, and illustration

Testing of the 2D stability of a polynomial by the method of this paper consists of Algorithm 3 and checking the stability conditions of Theorem 8. This section first outlines the proposed procedure to carry out the new test. Then its cost of computation is evaluated, and finally it is illustrated by a numerical example.

5.1. An outline for the stability test

Following is a possible outline with references to all the needed formulas for programming the test. For the sake of diversity, we shall use this time, instead of operations on polynomials, a more array-oriented presentation. In the following, $a * B$ denotes convolution of vector a by the vector B or, when B is a matrix, by each of the columns of B. B/a is used to denote deconvolution of (each of the columns of) B by a.

A test for determining whether $D(z_1, z_2)$ is stable (i.e., (2) holds for (1)) may proceed as follows.

(The "exit" presents points at which the procedure terminates with the verdict "not stable" because a necessary condition for stability is already found not to hold.)

a. Test whether $D(z_1, 1)$ is stable. If not stable - "exit". Test whether $D(1, z_2)$ is stable. If not stable - "exit". [Optionally, more univariate polynomials whose 1D stability is necessary for 2D stability may be tested, like $D(z_1, a)$ or $D(a, z_2)$ for any $|a| \geq 1$. If any is found not stable - "exit".]

b. Convert $D = [d_0, d_1, \ldots, d_{n_2}]$ into $Q = [q_0, q_1, \ldots, q_{2n_2}]$. To implement (7) proceed as follows. Obtain the columns of Q by the following sums of convolutions (see (51)):

$$q_k = \sum_{i=\max(0,k-n_2)}^{\min(k,n_2)} d_i * d^{\sharp}_{k-i}, \quad k = 0, \ldots, 2n_2.$$

Then to implement the conversion of Q (that has size $(2n_1+1) \times (2n_2+1)$ and symmetric columns) into $H = [h_0, \ldots, h_{n_2}]$ (whose size is $(n_1 + 1) \times (2n_2 + 1)$), follow the scheme worked out in the Appendix:

$$H = B_{n_1} \acute{I}_{n_1} Q_{(n_1:2n_1,0:2n_2)}. \tag{48}$$

Namely, premultiply the last $n_1 + 1$ rows of Q by $B_{n_1} \acute{I}_{n_1}$, where B_{n_1} is the Chebyshev matrix (59) that can be built using (62) and $\acute{I}$ is the $(n_1 + 1)$-th sized diagonal matrix diag$[1, 2, \ldots, 2]$.

c. Construct the 2D table using Algorithm 3. A reiteration of the algorithm in more array-oriented terms is as follows. Start with

$$R_0 = H + H^{\flat}, \quad R_1 = (H - H^{\flat})/[-1, 1]^t. \tag{49}$$

Then for $m = 0, 2, \dots, n-2$ ($n := 2n_2$), do:

$$[0\ R_{m+2}\ 0] = (r_{[m]\ 0} * ([R_{m+1}\ 0] + [0\ R_{m+1}]) - r_{[m+1]\ 0} * R_m)/\eta_m, \quad (50)$$

where $\eta_0 = 2, \eta_1 = 1$, and $\eta_m = r_{[m-1]\ 0}$ for $m \geq 2$ (where "0" denotes padding of the array by a column of zeros). [Optionally, more of the observed necessary conditions for stability $\rho_m(x) = R_m(x, 1) > 0$ and $r_{[m]0}(x) > 0$ for all $x \in [-1, 1]$ may also be tested. If any is contradicted - "exit"].

d. Test the polynomial $\rho(x) = \mathbf{x}^t R_n$ (of degree $2n_1n_2$) for the condition $\rho(x) \neq 0$ for all $x \in [-1, 1]$. (Equivalently, test the condition $\rho(x) > 0$ for all $x \in [-1, 1]$). If the condition does not hold "exit." If the condition holds (and the procedure reached this point without an earlier "exit") then $D(z_1, z_2)$ is stable.

The testing of the condition $\rho(x) \neq 0$ for all $x \in [-1, 1]$ can be carried out by the construction of a Sturm sequence and examination of the implied sign variation rules at $x = 1$ and $x = -1$, see, e.g., [1]. Alternatively, it is possible to transform $\rho(x) = \mathbf{x}^t \rho$ to the double degree symmetric polynomial $g(s)$ of (52) obtained by reversing the relations (61) and test the condition $g(s) \neq 0$ for all $s \in T$ using the extension of the underlying 1D stability test to the unit-circle zero location procedure in [3]. A new version for this procedure that handles more neatly singularities that may occur in this process has appeared recently in [10].

5.2. *Cost of computation*

Let $\ell_r(m)$ denote the degree of the variable x of $R_m(x, z)$. The first values are clearly $\ell_r(0) = n_1$, $\ell_r(1) = n_1$, $\ell_r(2) = 2n_1$, $\ell_r(3) = 3n_1$. From here on the $\ell_r(m)$ begin to obey the difference equation $\ell_r(m+3) - \ell_r(m+2) - \ell_r(m+1) + \ell_r(m) = 0$. Its solution, with the above initial conditions, is $\ell_r(m) = mn_1$ (for $m \geq 1$). The multiplication of two polynomials of degree k_1 and k_2 requires $(k_1 + 1)(k_2 + 2)$ flops. The division of a factor of degree k_2 from a polynomial of degree $k_1 + k_2$ requires $k_2(k_2 + 1)/2$ flops (plus k_2 divisions) assuming, as in the case here, $k_1 < k_2$. Summing the amount of computation at step m of the recursion over $m = 2, \dots, n := 2n_2$ and taking into account computation savings admitted by the column symmetries yields the figure $\frac{5}{3}n_1^2n_2^4 + O(n_{1,2}^5)$, where $O(n_{1,2}^k)$ is used to denote a bivariate polynomial in n_1 and n_2 with power terms $n_1^{k_1}n_2^{k_2}$ such that $k_1 + k_2 \leq k$. The derivation of the sequence $R_m(x, z)$ is the dominant part in the cost of the procedure. Step a requires $O(n_1^2)$ operations. The conversion of D to H in step b, as shown in the Appendix, requires $O(n_{1,2}^4)$ and step d requires again $O(n_{1,2}^4)$. Thus, the overall cost for this 2D stability test is $\frac{5}{3}n_1^2n_2^4 + O(n_{1,2}^5)$ real flops.

For comparison, the scattering-type tabular test of Hu and Jury in [16], [9] requires a somewhat higher cost of $\frac{5}{2}n_1^2n_2^4+O(n_{1,2}^5)$. The immittance-type tabular tests in [6] and [7], which stem from the form of BT for complex 1D polynomials, require $\frac{5}{6}n_1^2n_2^4+O(n_{1,2}^5)$, i.e., about half of the current cost. The current increase in cost is caused by the preliminary doubling of the polynomial second variable degree. Very interestingly, this doubling is compensated by the symmetry of the immittance approach, so that the current 2D tabular test still compares well with the most efficient 2D tabular test that can be developed from 1D stability tests that stem from any of the classical Schur-Cohn and Marden-Jury 1D stability tests.

5.3. Numerical example

For illustration, consider the following polynomial that appeared in several previous publications on 2D stability since the early days of this research topic (but note that we multiplied all coefficients by 4, and that some of the papers use other conventions that may reverse order of rows and/or columns in the matrix of coefficients):

$$D(z_1,z_2)=[\ 1\quad z_1^1\quad z_1^2\ ]\begin{bmatrix} 0 & 0 & 1\\ 0 & 1 & 2\\ 1 & 2 & 4\end{bmatrix}\begin{bmatrix} 1\\ z_2^1\\ z_2^2\end{bmatrix}.$$

Step (a): $D(z,1)=D(1,z)=[\ 1\quad 3\quad 7\ ]\mathbf{z}$ are easily determined to be stable.

Step (b). First convert D to

$$Q=\begin{bmatrix} 0 & 0 & 1 & 2 & 4\\ 0 & 1 & 4 & 9 & 10\\ 1 & 1 & 13 & 20 & 21\\ 0 & 1 & 4 & 9 & 10\\ 0 & 0 & 1 & 2 & 4\end{bmatrix}.$$

Then use (48) to convert Q into

$$H=\begin{bmatrix} 1 & 4 & 11 & 16 & 13\\ 0 & 2 & 8 & 18 & 20\\ 0 & 0 & 4 & 8 & 16\end{bmatrix}.$$

Apply to H (49)

$$R_0=\begin{bmatrix} 14 & 20 & 22 & 20 & 14\\ 20 & 20 & 160 & 20 & 20\\ 16 & 8 & 8 & 8 & 16\end{bmatrix},\quad R_1=\begin{bmatrix} 12 & 24 & 24 & 12\\ 20 & 36 & 36 & 20\\ 16 & 20 & 24 & 16\end{bmatrix}$$

and proceed with the recursion (50)

$$R_2 = \begin{bmatrix} 132 & 204 & 132 \\ 432 & 668 & 432 \\ 720 & 1056 & 720 \\ 608 & 848 & 608 \\ 256 & 320 & 256 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 864 & 864 \\ 4800 & 4800 \\ 12688 & 12688 \\ 19712 & 19712 \\ 19136 & 19136 \\ 11008 & 11008 \\ 3072 & 3072 \end{bmatrix}.$$

The first instance that a factor is divided out occurs in the creation of the next matrix R_4 (which for this low-degree example ends the sequence). R_4, obtained after dividing out the nontrivial factor $\eta_2(x) = r_{[1]\,0}(x) = [\ 12 \quad 20 \quad 16\ ]\mathbf{x}$, is

$$R_4 = \begin{bmatrix} 4320 & 30912 & 112208 & 257664 & 405632 & 444672 & 333056 & 156672 & 36864 \end{bmatrix}^t .$$

It can be checked that $\rho(x) = \mathbf{x}^t R_4$ has no zeros in $[-1, 1]$. Therefore, the tested bivariate polynomial is stable.

6. Conclusions

The paper presented a procedure to test the stability of a 2D discrete system polynomial by a tabular test that is based on the 1D test for real polynomials [3], [10]. It simplifies the stability conditions and reduces the overall complexity of related tests in [19], [20], [22] from a severe exponential order to a polynomial order of complexity. The drastic reduction in complexity has been achieved by exposition and elimination of certain redundant common factors in the earlier studies.

The proposed 2D stability test exhibits typical "immittance" formulation characteristics in that it uses a three-term recursion and propagates polynomials or matrices with a certain symmetry. This distinguishes it from the classical "scattering" approaches that use instead two-term recursions and propagate polynomials or matrices with no particular symmetry, as in [16], [9], and earlier less efficient tabular stability tests. However, the test here is also distinguishable from the immittance-type 2D tabular tests in [6] and [7], which, unlike the case here, stem from the complex polynomial form of the underlying 1D stability test [4], [10]. Notable differences include a longer sequence of matrices, matrices with column-symmetry instead of centro-symmetry, a different elimination rule of redundant factors, and more.

The count of operations for the current tabular stability test is higher by a factor of 2 than the cost of the immittance tabular tests in [6], [7]. The reason is the doubling of degree of the second variable that is needed in order to admit the use of a real polynomial 1D stability test. At the same time, its cost still compares well with the complexity of the most efficient scattering-type 2D tabular test, the tests in [16], [9] because the doubling of sequence length is approximately balanced by the computation of half the number of columns of each matrix that the symmetry of the immittance approach admits.

Recently, an approach called "telepolation" was used in [8], [11], and [9] to simplify the mentioned immittance and scattering 2D tabular tests that stem from complex 1D stability tests. The telepolation approach reduces the overall complexity of these tests from $O(n^6)$ to some very low $O(n^4)$ figures. It also shows that it is possible to carry out the 2D stability test by a finite collection of 1D stability tests. A similar simplification is also possible for the current tabular test, and it will be described in a forthcoming paper [13]. It will enlighten a distinctive advantage that the current tabular test introduces. All the other recent $O(n^4)$ tests that simplify 2D tabular tests that stem from complex 1D tests lead to testing 2D stability by a collection of univariate complex-coefficient polynomials. In contrast, the simplification of the current tabular test by telepolation admits a most efficient 2D stability test that involves real arithmetics only and uses a collection of 1D stability of only real polynomials.

Appendix: Conversion of $D(z_1, z_2)$ to $H(x, z)$

The test requires the conversion of the tested polynomial $D(z_1, z_2)$ to $Q(\tilde{s}, z)$ (7) followed by conversion of $Q(\tilde{s}, z)$ to $H(x, z)$ (9).

Denote the columns of D and Q by

$$D = [d_1, d_2, \ldots, d_{n_1}], \qquad Q = [q_0, \ldots, q_{2n_2}].$$

Write them as $D(s, z) = \mathbf{s}^t D\mathbf{z} = \sum_{i=0}^{n_2} d_i(s)z^i$ and $Q(s, z) = \mathbf{s}^t Q\mathbf{z} = \sum_{k=0}^{2n_2} q_k(s)z^i$, where each $d_i(s)$ is of degree n_1 and each $q_k(s)$ is of degree $2n_1$. It becomes apparent that the relation (7) implies that each q_k is obtained by the next sum of convolutions (with convolution denoted by $*$)

$$q_k = \sum_{i=0}^{n_2} d_i * d_{k-i}^{\sharp} = \sum_{i=\max(0,k-n_2)}^{\min(k,n_2)} d_i * d_{k-i}^{\sharp}, \quad k = 0, \ldots, 2n_2. \tag{51}$$

The range of the second sum reflects the fact that $d_i = 0$ for $i < 0$ and $i > n_2$.

Note that the computation of all the columns of Q requires $(n_2 + 1)^2$ convolutions. Each convolution requires $(n_1 + 1)^2$ flops. Thus, the cost of the conversion from D to Q is $(n_1 + 1)^2(n_2 + 1)^2$ flops.

Applying the "sharp" operation (i.e., reversion) to (51) leaves it unchanged. This proves that $q_k^{\sharp} = q_k$, i.e., $q_k(s)$ are all symmetric polynomials. A polynomial is symmetric if and only if its balanced polynomial assigns to it real values for all $s \in T$ (see, e.g., Theorem 3 in [4]). Therefore, $q_k(\tilde{s}) = s^{-n_1}q_k(s)$ are real for all $s \in T$. (However, the conversion of D to Q and the fact that the latter has symmetric columns do not depend on interpreting s as taking values on T.)

The second step involves the conversion of each symmetric polynomial $q_k(s)$

of degree $2n_2$ into $h_k(x) = \mathbf{x}^t h_k$ of degree n_1, where $H = [h_0, h_1, \ldots, h_{2n_2}]$ such that $h_k(x)\big|_{x=\frac{1}{2}(s^{-1}+s)} = q_k(\tilde{s})$.

The key problem is therefore the conversion of a symmetric polynomial of degree $2M$, say $g(s) = \sum_{i=0}^{2M} g_i s^i$ = with $g_{M-i} = g_{M+i}$, $i = 0, \ldots, M-1$, i.e., $g(\tilde{s}) = [s^{-M}, \ldots, 1, \ldots, s^M][g_0, \ldots, g_M, \ldots, g_0]^t$ into another polynomial of half degree, say $\rho(x) = \sum_{i=0}^{M} \rho_i x^i$, such that the two polynomials are related by

$$\rho(x)\big|_{x=\frac{1}{2}(s^{-1}+s)} = g(\tilde{s}). \tag{52}$$

For later convenience, we express the latter requirement as

$$\begin{aligned} [1, x, \ldots, x^M][\rho_0, \rho_1, \ldots, \rho_M]^t\big|_{x=\frac{1}{2}(s^{-1}+s)} &= [s^{-M}, \ldots, 1, \ldots, s^M] \\ &\quad [g_0, \ldots, g_M, \ldots, g_0]^t. \end{aligned} \tag{53}$$

This problem has been treated before via trigonometric identities for the conversion of $D(z_1, z_2)$ to $H(x, z)$ in [14] and even before (for replacing the testing of $g(s)$ for no zeros on T by testing $\rho(x)$ for no zeros on $[-1, 1]$) in [1]. In what follows, it will become apparent that the assumption in these previous derivations that s and x take values on T and $[-1, 1]$, respectively, is superfluous.

Define, as in [2], balanced polynomials of the following specific forms:

$$G_m(s) = \frac{s^{-m} + s^m}{2}, \quad m = 0, 1, 2, \ldots. \tag{54}$$

It is easily verified that they obey the recursion

$$G_{m+1}(s) = (s^{-1} + s)G_m(s) - G_{m-1}(s), \quad m \geq 1. \tag{55}$$

Starting this relation with $G_0(s) = 1$ and $G_1(s) = \frac{s^{-1}+s^1}{2}$ can be used to generate the $G_m(s)$ recursively. By induction, it follows that the transformation (8) matches to each $G_m(s)$ a $T_m(x)$ given by $T_0(x) = 1T_1(x) = x$ and afterwards obtainable by the recursion

$$T_{m+1}(x) = 2xT_m(x) - T_{m-1}(x). \tag{56}$$

The latter recursion with the indicated initiations is recognized as the generating recursion for the familiar (first kind) Chebyshev polynomials. Thus, we showed that

$$T_m(x)\big|_{x=\frac{1}{2}(s^{-1}+s)} = G_m(s). \tag{57}$$

The Chebyshev polynomials are formally defined over $x \in [-1, 1]$ by $T_m(x) = \cos(n \cos^{-1}(x))$, and they have a variety of valuable mathematical properties, none of which are of concern here. In fact, we stress that the relation (57) between $G_m(s)$ and $T_m(x)$ does not rely on assuming $s \in T$ or $x \in [-1, 1]$ but was derived here directly from their respective generating recursions.

Develop $g(\tilde{s})$ into

$$\begin{aligned} g(\tilde{s}) &= [s^{-M}, \dots, s^{-1}, 1, s, \dots, s^{M}][g_0, \dots, g_{M-1}, g_M, g_{M-1}, \dots, g_0]^t \\ &= [1, \frac{s^{-1}+s}{2}, \dots, \frac{s^{-M}+s^{M}}{2}][g_M, 2g_{M-1}, \dots, 2g_0]^t \\ &= [G_0(s), G_1(s), \dots, G_M(s)]\acute{I}_M[g_M, g_{M-1}, \dots, g_0]^t, \end{aligned} \tag{58}$$

where $\acute{I}_M := \operatorname{diag}[1, 2, \dots, 2]$. Next, denote the coefficients of $T_m(x)$ by $T_m(x) = \sum_{i=0}^{m} t_{m,i}x^i = \mathbf{x}^t T_m$ and define B_M to be an $(M+1)$–sized upper triangular square matrix whose columns are the coefficients $b_m = [t_{m,0}, \dots, t_{m,m}, 0, \dots, 0]^t$ of the Chebyshev polynomials (padded by zeros to length $M+1$), viz.,

$$B_M = \left[\begin{array}{c|c|c|c|c} T_0 & T_1 & \multirow{2}{*}{$\cdots$} & T_{M-1} & \multirow{2}{*}{T_M} \\ 0_M & 0_{M-1} & & 0 & \end{array}\right]. \tag{59}$$

(0_m is a vector of zeros of length m.) It follows that

$$[1, x, \dots, x^{M}]B_M = [T_0(x), T_1(x), \dots, T_M(x)]. \tag{60}$$

To evaluate (58), use (57) and (60) and compare the outcome with (52). One obtains

$$\begin{aligned} g(\tilde{s}) &= [T_0(x), \dots, T_M(x)]\acute{I}_M[g_M, \dots, g_0]^t \\ &= [1, x, \dots, x^{M}]B_M\acute{I}_M[g_M, \dots, g_0]^t = \rho(x). \end{aligned}$$

This proves that the coefficients of $\rho(x)$ and $g(s)$, see (53), are related by

$$\begin{bmatrix} \rho_0 \\ \rho_1 \\ \vdots \\ \rho_M \end{bmatrix} = B_M\acute{I}_M \begin{bmatrix} g_M \\ g_{M-1} \\ \vdots \\ g_0 \end{bmatrix}. \tag{61}$$

The recursion (56) induces a recursive algorithm to obtain the columns of the matrix B_M (59) required in (61) as follows. Starting with $T_0 = 1$ and $T_1 = [0, 1]^t$ for $m = 1, \dots, M-1$, do:

$$T_{m+1} = 2\begin{bmatrix} 0 \\ T_m \end{bmatrix} - \begin{bmatrix} T_{m-1} \\ 0 \end{bmatrix}. \tag{62}$$

To summarize, the conversion of Q to H can be obtained by multiplying the submatrix formed by the last n_1+1 rows of Q by $B_{n_1}\acute{I}_{n_1}$, as depicted in (48). Exploiting the fact that B_{n_1} is an upper triangular matrix with alternatingly zero entries, this conversion requires approximately $\frac{1}{4}n_1(n_1+1)(n_2+1)$.

References

[1] B. D. O. Anderson and E. I. Jury, Stability test for two-dimensional recursive filters, *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 366–372, Aug. 1973.

[2] Y. Bistritz, A discrete stability equation theorem and method of stable model reduction, *Systems Control Lett.*, vol. 1, pp. 373–381, 1982.

[3] 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.

[4] Y. Bistritz, A circular stability test for general polynomials, *Systems Control Lett.*, vol. 7, no. 2, pp. 89–97, 1986.

[5] Y. Bistritz, An immittance-type stability test for two-dimensional digital filters, *Proceedings of the 28th Annual Asilomar Conference on Signals, Systems, and Computers*, pp. 918–922, 1994.

[6] Y. Bistritz, Stability testing of two-dimensional discrete linear system polynomials by a two-dimensional tabular form, *IEEE Trans. Circuits and Systems, Part I*, vol. 46, pp. 666–676, June 1999.

[7] Y. Bistritz, Immittance-type tabular stability test for 2-D LSI systems based on a zero location test for 1-D complex polynomials, *Circuits Systems Signal Process*. vol. 19, pp. 245–265, 2000.

[8] Y. Bistritz, Stability testing of 2-D discrete linear systems by telepolation of an immittance-type tabular test, *IEEE Trans. Circuits and System, Part I*, vol. 48, pp. 840–846, July 2001.

[9] Y. Bistritz, Stability testing of two-dimensional discrete-time systems by a scattering-type tabular form and its telepolation, *Multidimensional Systems Signal Process.*, vol. 13, pp. 55–77, Jan. 2002.

[10] Y. Bistritz, Zero location of polynomials with respect to the unit-circle unhampered by nonessential singularities, *IEEE Trans. Circuits and Systems Part I* , vol. 49, pp. 305–314, Mar. 2002.

[11] Y. Bistritz, On testing stability of 2-D discrete sytems by a finite collection of 1-D stability tests for complex polynomials, *IEEE Trans. Circuits and Systems Part I*, vol. 49, pp. 1634–1638, Nov. 2002.

[12] Y. Bistritz, An efficient integer-preserving stability test for discrete-time systems, submitted.

[13] Y. Bistritz, Testing stability of 2-D discrete systems by a set of real 1-D polynomials, submitted.

[14] N. K. Bose, Implementation of a new stability test for two-dimensional filters, *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-25, pp. 117–120, 1977.

[15] N. K. Bose, *Applied Multidimensional System Theory*, Chap. 3, Van Nostrand Reinhold, Princeton, NJ. 1982.

[16] X. Hu and E. I. Jury, On two-dimensional filter stability test, *IEEE Trans. Circuits and Systems* vol. 41, pp. 457–462, July 1994.

[17] E. I. Jury, Stability of multidimensional systems and related problems, in *Multidimensional Systems: Techniques and Applications*, S. G. Tzafestas, ed., Marcel Dekker, New York 1986.

[18] E. I. Jury, Modified stability table for 2-D digital filter, *IEEE Trans. Circuits and Systems* vol. CAS-35, 116–119, 1988.

[19] B. M. Karan and M. C. Sarisvastava, A new test for 2-D filters, *IEEE Trans. Circuits and Systems*, vol. CAS-33, pp. 807–809, 1986.

[20] B. M. Karan and M. C. Sarisvastava, Correction to A new test for 2-D filters, *IEEE Trans. Circuits and Systems*, vol. CAS–36, p. 167, 1989.

[21] B. T. O'Connor and T. S. Huang, Stability of general two-dimensional recursive filters, in *Two-dimensional Digital Signal Processing I: Linear Filters*, (T. S. Huang, ed.,) Springer, Berlin, 1981.

[22] K. Premaratne, Stability determination of two-dimensional discrete-time systems, *Multidimensional Systems Signal Process.*, vol. 4, no. 4, pp. 331–354, 1993.

[23] K. Premaratne and E. I. Jury, On the Bistritz tabular form and its relationship with the Schur-Cohn minors and inner determinants, *J. Franklin Institute*, vol. 330, no. 1, pp. 165–182, 1993.

[24] D. D. Siljak, Stability criteria for two-variable polynomials, *IEEE Trans. Circuits and Systems*, vol. CAS-22, pp. 185–189, Mar.1975.