Critical Stability Constraints for Discrete-Time Linear Systems

Yuval Bistritz, *Fellow, IEEE*

***Abstract*—Critical stability constraints are a small set of conditions that are enough to maintain the stability of a system when some parameters are perturbed from a nominal stable setting. The paper uses a recently introduced efficient integer-preserving (IP) form of the Bistritz test to derive critical constraints for stability of discrete-time linear systems. The new procedure produces polynomial (rather than rational) constraints of particularly low degree whose variates are the free parameters (or the literal coefficients) of the system's characteristic polynomial. Comparison with the modified Jury test, also an efficient IP stability test, shows that the constraints are obtained with less computation and, more contributing to simplicity, the constraints appear as polynomials with degrees lower by a factor of two.**

***Index Terms*—Discrete-time systems, immittance algorithms, integer-preserving (IP) computation, modified Jury test (MJT), stability constraints, stability test.**

I. Introduction

ALGEBRAIC stability tests are methods to determine stability of a linear system without using numerical means like calculation of zeros. One of their use is to obtain stability constraints for some literal coefficients of the characteristic polynomial during the design of filter or a control for a system. Typically, the number of constraints necessary and sufficient for stability is equal to the order of the system.

Critical stability constraints are a simpler set of conditions that can help a designer to determine the flexibility that he has in changing parameters in an already stable system without loosing its stability. In such a situation, knowing a nominal stable state reduces the number of constraints that must be attended. In fact, the lower number of constraints becomes a fixed number (three and even less) that is independent of the order of the system. It should be noted though that the complexity of the constraints does raise with the system's order.

This paper focuses on critical stability constraints for discrete-time systems. In this case, one is given the characteristic polynomial with one or more undetermined parameters (e.g., some of its coefficients are literal). It is also known that the polynomial is "stable" (has all its zeros inside the unit-circle of the complex plane) for a nominal set of coefficients. And the critical stability constraints should contain (in as "simple" a setting as possible) the answer to the question: to what extent the variable coefficients may be perturbed without causing any of the zeros to move outside the unit-circle.

Critical stability constraints for linear systems (both continuous and discrete) were considered by Jury in [1], [2]. The critical conditions for discrete systems were initially posed on determinants of Schur–Cohn and Inner matrices and he demonstrated their calculation for low degree polynomials ($n = 3, 4, 5$) [1], [2]. In order to derive stability constraints for a polynomial of any degree in a systematic way, one needs an algebraic stability test (i.e., a recursive algorithms often also called a "tabular" test because traditionally they were presented in a tabular format).

In principle, any decent algebraic stability test can be used to obtain stability constraints for some undetermined parameters. The arsenal of stability tests include the classical tests of Schur–Cohn and the Marden–Jury tests that appeared in a variety of version and were surveyed and classified in [3]. A more recent addendum to the above pack is the Bistritz test (BT) [4], [5]. It differs from all the above tests in the form of recursion (three-term recursion of symmetric polynomials as opposed to two-term recursion of polynomials with no symmetry) and in efficiency (about half of the number of arithmetic operations compared to the most efficient test in Schur–Cohn Marden–Jury class). The collective term "immittance" (also "split") is often used to distinguish various algorithms that stem from the BT formulation from their so called "scattering" classical counterpart.

Trying to use some of the above tests to determine an intervals of stability for one free parameter (we shall run such simple examples later) is enough to convey the feeling that not all tests are equally well behaving for the task. This raises the question, that was not truly addressed before, what makes one stability test more suitable than another for deriving critical stability constraints. This question depends on another one – what considerations count in determining the relative efficiency of different stability tests for this task. This paper is about raising this question, postulating reasonable measures, and offering a new solution that is a best performer according to the postulated requirements.

The paper will present critical stability constraints for discrete-time linear systems based on an integer-preserving (IP) form of the BT in [6]. It will be argued and illustrated that integer preservation is the most desirable property in a stability test in order to deal with literal coefficients. Next, it will also be demonstrated that the most adequate measure for simplicity of an IP test for this task is having an algorithm with as low as possible growth of the length of the integers. By these criteria, the best alternative scattering-type stability test available for critical stability is the modified Jury test (MJT), a test that Jury presented in several occasions and versions, including [7]–[9] (see the "C-type" category in [3]) and the main reason for this is the fact that it is too an efficient IP test [10].

Jury too maintained that the MJT is favorable for critical stability constraints but he reached this conclusion from a different perspective. He designed this test to produce iteratively the sequence of determinants of principal submatrices of the

Manuscript received March 11, 2005. This paper was recommended by Associate Editor P. Bauer.

The author is with Department of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: bistritz@eng.tau.ac.il).

Digital Object Identifier 10.1109/TCSII.2005.856032

Schur–Cohn–Fujiwara (SCF) matrix (that in turn are also equal to respective Inner determinants). As a consequence, they provide an immediate algorithm to obtain the critical constraints that he previously posed on the SCF principal minors. Premaratne and Jury studied the relation of the BT to the SCF principal minors in [11] and subsequently considered in [12] the use of the BT and the MJT for critical stability constraints. They count in [12] pros and cons for both the BT and the MJT without a conclusive choice between the two. This paper will obtain critical stability constraints that (by the proposed measures) are simpler and are obtained more efficiently than using the original BT or the MJT.

II. New Procedure for Stability Constraints

A. Stability Constraints

Assume a real polynomial

$$D_n(z)=\sum_{i=0}^{n} d_i z^i,\ D_n(1)\neq 0,\quad d_n>0. \tag{1}$$

Let the reciprocal of a polynomial $D_n(z)$ be defined and denoted by $D_n^\sharp(z):=\sum_{i=0}^{n} d_{n-i}z^i$. A polynomial is called symmetric if $D_n(z)=D_n^\sharp(z)$. A symmetric polynomial has coefficients that exhibit symmetry with respect to their center, i.e., $d_{n-i}=d_i,\ i=0,\ldots,n$.

The following stability test for real polynomials was presented in [6].

Algorithm 1: Assume $D_n(z)$ (1) and assign to it a sequence of $n+1$ symmetric polynomials $\{R_m(z)=\sum_{k=0}^{n-m} r_{m,k}z^k,\quad m=n,\ \ldots,0\}$, as follows:

$$R_n(z)=D_n(z)+D_n^\sharp(z),\quad R_{n-1}(z)=\frac{D_n(z)-D_n^\sharp(z)}{(z-1)}. \tag{2}$$

Set $\eta_n=2$, $\eta_{n-1}=1$ and for $m=n-1,n-2,\ldots,1$ do

$$zR_{m-1}(z)=\frac{r_{m+1,0}(z+1)R_m(z)-r_{m,0}R_{m+1}(z)}{\eta_{m+1}}$$
$$\eta_{m-1}=r_{m,0}. \tag{3}$$

Theorem 1: The polynomial $D_n(z)$ (1) is stable if, and only if, $r_{n-1,0}\ (=d_n-d_0)>0$ and Algorithm 1 yields for it

$$R_m(1)>0,\quad m=n,n-1,\ldots,0. \tag{4}$$

If $D_n(z)$ is stable then the next conditions hold as well

$$r_{m,0}>0,\quad m=n,n-1,\ldots,0. \tag{5}$$

Normal conditions are defined as the case when all the polynomials produced by Algorithm 1 are with nonzero leading coefficient $r_{m,m}\neq 0$. The second part of Theorem 1 implies that normal condition are necessary conditions for stability (noting that $r_{m,0}=r_{m,m}$).

Let $\mathbb{I}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the ring of integers[1] and the fields of rational and real numbers, respectively, and $\mathbb{I}[z]$, $\mathbb{Q}[z]$, $\mathbb{R}[z]$ denote the set of polynomials with coefficients in the respective domains. It was shown in [6] that Algorithm 1 is IP. Namely, it has the property that if $D_n(z)\in\mathbb{I}[z]$ then $R_m(z)\in\mathbb{I}[z]$ for all $m=n,n-1,\ldots,0$. In this it differs from the original BT that assigns in general to $D_n(z)\in\mathbb{I}[z]$ a sequence of symmetric polynomials that are in $\mathbb{Q}[z]$ (except to the first two). Algorithm 1 remains fraction-free even though it contains divisions because the divisor $\eta_{m+1}=r_{m+2,0}$ is a common factor to all the integer coefficients of the polynomial in the numerator of (3). As a consequence, it will be seen that derivation of stability constraints for polynomials with literal coefficients with the above test leads to a set of polynomial inequalities as opposed to a set of rational functions inequalities when using the original form of the BT.

Furthermore, Algorithm 1 is an IP stability test with a very restrained growth of coefficients length. To state this feature length of an integer and a measure for length of coefficients in an integer polynomial need to be defined (though not too rigorously as we are looking for mostly for a qualitative characterization). The length of an integer can be measured by decimal-size or its bit-size (or maybe by $\log$ on base 10 or 2 of the absolute integer). Let then B characterize the length of coefficients of $D_n(z)\in\mathbb{I}[z]$ (say, B is the length of the longest coefficient of the polynomial). For further simplicity, Theorem 2 below assumes that B characterizes also the lengths of $R_n(z)$, $R_{n-1}(z)$.[2] With these assumptions, the following characterization was shown for Algorithm 1 in [6].

Theorem 2: Let B present a common bound for the coefficient length of $D_n(z)\in\mathbb{I}[z]$, $R_n(z)$ and $R_{n-1}(z)$ (as explained above). Then, the length of the coefficients of $R_{n-m}(z)\in\mathbb{I}[z]$ is bounded by mB, $m=2,\ldots,n$.

Accordingly, the longest coefficient that the algorithm produces for a polynomial bounded by B is bounded by nB. To appreciate how remarkable is this feature, it is also shown in [6] that taking the naive approach to reach integer preservation (by simply avoiding divisions) leads to a stability test with exponential growth of coefficients.

Example: Suppose one wants to determine for the following polynomial:

$$D_7(z;K)=K+3z+2z^2+4z^3+8z^4+7z^5+5z^6+8z^7 \tag{6}$$

all values of K (a real parameter) for which it is stable. For brevity we skip the polynomials that Algorithm 1 creates and only bring the resulting stability constraints required in Theorem 1. They are given by the next set of inequalities

$$\begin{aligned}
r_{6,0}(K)&=8-K>0\\
R_7(1;K)&=74+2K>0, R_6(1;K)=85-7K>0\\
R_5(1;K)&=384+58K-6K^2>0\\
R_4(1;K)&=2744+15K-92K^2+5K^3>0\\
R_3(1;K)&=6704+1382K-356K^2-54K^3+4K^4>0\\
R_2(1;K)&=22424-3309K-2792K^2+104K^3\\
&\quad+56K^4-3K^5>0\\
R_1(1;K)&=49760-2836K-12\,204K^2-1778K^3\\
&\quad+230K^4+30K^5-2K^6>0\\
R_0(1;K)&=\frac{(7-K)R_1(1)}{2}>0.
\end{aligned} \tag{7}$$

[1]The notation $\mathbb{Z}$ (instead of $\mathbb{I}$ that was used also in [6]) is maybe a more widely used notation for integers.

[2]If the length of $D_n(z)$ is bounded by B then it would be more correct to say that the lengths of $R_n(z)$ and $R_{n-1}(z)$ are bounded by $B+1$ and $B+2$, respectively (as implied by one and two passes, respectively, of additive operations). The implied (qualitatively not meaningful) correction can be easily obtained. However, the assumption that B also characterizes the first two polynomials in the sequence is exact for the following application of the bounds stated in Theorem 2 to feature also the growth of the degree of literal coefficients in constraints expressed as polynomials.

A solution to this set of inequalities necessitates numerical means. It can be done by finding the zeros or by plotting the polynomials as function of K. It is enough to consider only real zeros and intervals between them at which the polynomials are positive. The solution is given by the common intersection of all intervals at which the polynomials are positive. It is $-3.812 < K < 1.758$ (approximately) [1].

The way $R_0(1;K)$ is factored above is not coincidental. It follows from a useful general property. Namely, for $D_n(z) \in \mathbb{I}[z]$, $R_1(1)/2 \in \mathbb{I}$ is a factor of $R_0(1) \in \mathbb{I}[z]$. This is so because

$$R_1(1) = 2r_{1,0}, \quad r_{0,0} = \frac{r_{1,0}(2r_{2,0} - r_{2,1})}{r_{3,0}}. \tag{8}$$

Strictly speaking, Theorem 2 relates to integers. For example, suppose the above example is run with K = 1. Measuring the length of the tested polynomial by $B = \log_2(8) = 3$ then it is noticed that indeed the largest number, that is the last $r_{0,0} = 99\,600$ has length $\log_2(99600)$ below $nB = 21$. However, the underlying algorithm preserves coefficients over other integral (integer-like) domains. For example, the coefficients may also be polynomials in secondary variable(s) [13]. Thus, it is possible to regard Theorem 2 as characterizing the growth of also literal parameters in a polynomial, like K in the above numerical example, measuring the length for K in polynomial-coefficients by its highest power in it. In the above example, the length of K in $D_7(z)$ is $B = 1$. In this case, the power of K in the polynomials $R_m(z;k)$ (that were omitted for brevity) attains the length growth depicted by Theorem 2 at each step till it ends with $nB = 7$. Thus, the growth of coefficients provides a useful characterization for the complexity of an IP stability test as an algorithm to determine stability constraints on literal coefficients. The situation becomes more complicated when the number of literal coefficients is higher than one because in this case the constraints become multivariate polynomials. Theorem 2 still offers a bound on the highest power of each parameter in the polynomial constraints. The number of polynomials in the set of inequalities increase with the order of the tested system. However, here at least, the situation can somewhat be alleviated because in many a practical situation it may suffice to attend to only 3 or 2 (and even 1) critical constraints—our next topic.

B. Critical Constraints

Consider a polynomial $D_n(z;k) \in \mathbb{R}[z]$ with coefficients dependent on k, where "k" stands for one or several parameters.

$$D_n(z;k) = \sum_{i=0}^{n} d_i(k) z^i \in \mathbb{R}[z]. \tag{9}$$

Assume that it is known that the polynomial $D_n(z;k)$ is stable for a nominal value $k = k_o$ of the parameter(s), viz.,

$$D(z;k_o) \neq 0 \quad \forall |z| \geq 1; \quad d_n(k_o) > 0 \tag{10}$$

where we added here the assumption $d_n(k_o) > 0$ (with no loss of generality because else $-D(z;k)$ can be considered). Critical stability constraints aim to find a small number of condition that are sufficient to determine the largest vicinity V_{k_o} of $k_o \in V_{k_o}$ such that $D(z;k)$ is stable $\forall k \in V_{k_o}$. (We say "small" and not "minimal" because it will become apparent that the minimal is not necessarily the best choice.)

The derivation of critical stability constraints relies on the continuous variation of the location of zeros of a polynomial as function of continuous variation of its coefficients. Given that at k_o the polynomial has all its zeros inside the unit-circle, they remain there as k varies away from k_o till zeros (one or more) touch the unit-circle. The role of critical stability constraints is to detect when the root locus touches the unit-circle in the above described manner. Therefore, to proceed we need to know the behavior of Algorithm 1 at the presence of unit-circle zeros. The following lemma can be proved in much the same way as its counterpart in the original test form [4, Th. 4.2] (or [5, Th. 2]). In the following "UC zeros" mean zeros on the unit-circle and "RP zeros" are reciprocal pairs of zeros with respect to it, (z_r, z_r^{-1}), $|z_r| \neq 1$.

Lemma 1: If the recursion (3) is interrupted by a $R_{s-1}(z) \equiv 0$ (assuming normal conditions before) then $R_s(z)$ contains all the UC and RP zeros of $D_n(z)$. And conversely, if the total number of UC and RP zeros of $D_n(z)$ is s, then the above recursion will be interrupted by a $R_{s-1}(z) \equiv 0$ (assuming normal conditions till then).

The reservation about normal conditions in the above lemma is made because Algorithm 1 needs to be amended in case it faces abnormal polynomials (with vanishing leading coefficients). However, the lemma and the algorithm are broad enough to our current needs.

Theorem 3: Given (10), the largest vicinity V_{k_o} of stability of $D(z;k)$ obeys, and can be determined by, either of the following three sets of critical conditions:

1) $D_n(1;k) > 0$, $(-1)^n D_n(-1;k) > 0$, $r_{1,0}(k) > 0$;
2) $(-1)^n D_n(-1;k) > 0$, $r_{1,0}(k) > 0$;
3) $r_{0,0}(k) > 0$.

Here, $r_{m,i}(k)$ are obtained for $D(z;k)$ by Algorithm 1.

Proof: Since $D(z;k_o)$ is stable, Algorithm 1 is normal for $D(z;k_o)$. In particular $R_s(z;k_o) \equiv 0$, for $s \geq 0$, can not occur. Consider the location of zeros of $D(z;k)$ as k varies away from k_o. At k_o all zeros are inside the unit-circle. Stability breaks either when one or more real unit-circle zero (with or without multiplicity) or when one or more complex conjugate pair of zeros (with or without multiplicity) migrate to the unit-circle. According to Lemma 1, both situations imply a vanishing symmetric polynomial. In this situation the converse is also true. Namely, a vanishing polynomial can not imply in our case RP zeros (because for k_o there are no zeros outside the unit-circle).

Real zeros on the unit-circle can occur either at $z = 1$ or at $z = -1$. Let $D_n(z;k) = d_n(k)\prod_{i(k)=1}^{n}(z - z_{i(k)})$. Since $d_n(k_o) > 0$, $D_n(1;k_o) = d_n(k_o)\prod_{i(k_o)=1}^{n}(1 - z_{i(k_o)}) > 0$. Thus, as long as $D_n(1;k) > 0$ no zero(s) at $z = 1$ can occur as we move away from $k = k_o$. For $z = -1$, $D_n(-1;k_o) = d_n(k_o)\prod_{i(k_o)=1}^{n}(-1 - z_{i(k_o)})$ and $d_n(k_o) > 0$ implies that as long as $(-1)^n D_n(-1;k) > 0$ zeros can not break away from the unit-circle also at $z = -1$. Assume next the case of complex zeros. Namely, assume that the zero location of the polynomial $D_n(z;k)$ first crosses the unit-circle for some $\hat{k}$ with one (or more) complex-conjugate pair of zeros. Say the multiplicity of this pair (or one of several distinct pairs) is $\ell \geq 1$. It then will nullify $R(z;\hat{k}) \equiv 0$ for $i = 2\ell - 1, \ldots, 1$. Thus, irrespective of its multiplicity, it will be detected (also) by $R_1(z;\hat{k}) \equiv 0$. Since $R_1(z;k) = r_{1,0}(k)(z+1)$. $R_1(z;\hat{k}) \equiv 0$ is equivalent to

$R_1(1;\hat{k})=0$ and also to $r_{1,0}(\hat{k})=0$. This completes the proof that 1) is an adequate set of critical constraints.

To prove 2) we need to show that $D_n(1;k)>0$ in 1) is superfluous. A $D_n(1,\hat{k})=0$ implies $R_m(1;\hat{k})=0$ for all m (as realized by setting $z=1$ into (3)). Therefore, in particular, $R_1(1;\hat{k})=2r_{1,0}(\hat{k})=0$. Namely, zeros at $z=1$ are detected by the second constraint in 2).

It remains to prove that the single constraint in 3) is also enough. $r_{1,0}(\hat{k})=0$ implies $r_{0,0}(\hat{k})=0$, as can be realized from (8). Therefore, $r_{0,0}(k)>0$ may replace $r_{1,0}(k)>0$. $D_n(-1,\hat{k})=0$ implies $R_m(-1;\hat{k})=0$ for all m. Therefore, such a zero is detected by any $R_m(-1;\hat{k})=0$ for m even. (Note that the odd degree polynomials in Algorithm 1 vanish always at $z=-1$ by being symmetric polynomial.) Thus, in particular, $r_{(0,0)}(k)>0$ covers also the constraint $(-1)^n D_n(-1;k)>0$. ■

Set 2) is better than set 1) simply because it omits one of the conditions in 1). Set 3) looks even more attractive in its mathematical conciseness. But, it is not necessarily more useful, for at least the problem at hand here. (Obtaining $r_{0,0}(k)$ requires an extra step of the algorithm creating a higher degree polynomial that is anyway known to be factorable into $r_{1,0}$ and $D_n(-1;k)$.)

Example: To illustrate the critical constraints in Theorem 3, consider again the polynomial (6) and assume that we know that it is stable at $K=1$ (say). Then, according to set 1) we have to examine

$$D_7(1;K)=K+37>0, \qquad -D_7(-1;K)=K+7>0 \tag{11}$$

and $r_{1,0}(K)=R_1(1;K)/2$ depicted in Example 1. According to 2), $D_7(1;K)>0$ can be dropped of the set. Finally, as illustration for the argument used to prove 3), it is noticed that the last constraint in (7), $r_{0,0}(K)=(7-K)r_{1,0}(K)>0$, indeed covers the two constraint $r_{1,0}(K)>0$ and $(-1)^7D_7(-1;K)>0$.

III. Stability Constraints With MJT

Here, we bring a brief account on the use of the MJT to determine critical stability constraints. We shall use a more loose presentation because most of the things that we said here can be found in one or more of the references that were cited.

As already mentioned, Jury presented on several occasions (and several versions), including [7]–[9], a modified form of the Marden–Jury test that aims to produces explicitly the principal minors of the Schur–Cohn matrix. The next presentation follows the C-prototype form in [3] where the relation between the various MJT tests and their relation to the Schur–Cohn principal minors are rigorously proved.

Algorithm 2: Assign to $D_n(z)$ (1) a sequence of $n+1$ polynomials $\{C_m(z)=\sum_{i=0}^{m}c_{m,i}z^i, \quad m=n-1,\ldots,0\}$, as follows:

$$zC_{n-1}(z)=d_nD(z)-d_0D^\sharp(z), \quad q_{n-1}=1. \tag{12}$$

For $m=n-1,\ldots,1$ do

$$zC_{m-1}(z)=\frac{c_{m,m}C_m(z)-c_{m,0}C_m^\sharp(z)}{q_m}, \quad q_{m-1}=c_{m,m}. \tag{13}$$

Theorem 4: $D_n(z)$ is stable if, and only if, $c_{m,m}>0$ for all $m=1,\ldots,n$.

It can also be shown (see [3] for details) that the leading coefficients $c_{m,m}, m=n-1,\ldots,0$ of the polynomials that Algorithm 2 produces are the principal minors of the SCF matrix, an $n\times n$ symmetric matrix that can be built for $D_n(z)$ such that its positive definiteness is necessary and sufficient for stability. More pertinent to the current application, than its relation to the principal minors of the SCF matrix, is the fact that the MJT is IP, as was proved in [10]. Namely, if $D_n(z)\in\mathbb{I}[z]$ then $C_m(z)\in\mathbb{I}[z]$, $m=n-1,\ldots,0$. Let B presents the length of $D_n(z)$ then the length of the coefficients of $C_{n-m}(z)$ is bounded by $2mB, m=1,\ldots,n$. This property was too stated in [10]. It is easily verified as follows. Let $\ell_c(m)$ denote the coefficient length measure for $C_{n-i}(z)$. We have $\ell_c(0)=B$ (with $C_n(z)=D_n(z)$), $\ell_c(1)=2B$ and $\ell_c(2)=2B$ (no division yet) then for $m\geq 2$, the recursion includes division and it follows that $\ell_c(m+1)=2\ell_c(m)-\ell_c(m-1)$. The solution of this difference equation for the given initial conditions is $\ell_c(m)=2mB$. Thus, coefficients in Algorithm 2 grow at twice the rate of respective degree polynomials in Algorithm 1, cf. Theorem 2.

Example: Applying Algorithm 2 to our running example (6) gives according to Theorem 4 the following constraints as necessary and sufficient for stability:

$$\begin{aligned}
c_{6,6}(K)&=64-K^2>0\\
c_{5,5}(K)&=3520+240K-153K^2+K^4>0\\
c_{4,4}(K)&=193\,536+26\,752K-13\,328K^2-914K^3\\
&\quad+291K^4-K^6>0\\
c_{3,3}(K)&=10\,261\,568+2\,274\,320K-913\,263K^2-150\,236K^3\\
&\quad+32\,867K^4+2596K^5-493K^6+K^8>0\\
c_{2,2}(K)&=335\,643\,968+66\,494\,928K-54\,317\,131K^2\\
&\quad-10\,539\,062K^3+2\,688\,419K^4+423\,308K^5\\
&\quad-63\,886K^6-5254K^7+711K^8-K^{10}>0\\
c_{1,1}(K)&=7\,366\,072\,320-167\,037\,952K-2\,512\,155\,808K^2\\
&\quad-297\,274\,856K^3+209\,936\,218K^4\\
&\quad+31\,904\,728K^5-6\,550\,631K^6\\
&\quad-914\,104K^7+108\,497K^8+8520K^9\\
&\quad-933K^{10}-12\,848K^{11}+K^{12}>0\\
c_{0,0}(K)&=160\,324\,729\,600-36\,845\,389\,120K\\
&\quad-76\,623\,046\,164K^2+2\,144\,079\,248K^3\\
&\quad+12\,888\,369\,908K^4+1\,581\,421\,672K^5\\
&\quad-566\,275\,189K^6-89\,742\,192K^7+11\,924\,840K^8\\
&\quad+1\,775\,976K^9-156\,894K^{10}-12\,848K^{11}\\
&\quad+1164K^{12}-K^{14}>0.
\end{aligned} \tag{14}$$

It is notable that the degrees of the above polynomials follow, as argued before for also Algorithm 1, the growth of coefficients size, that this time is ruled by $2mB, m=1,\ldots,n$. The MJT can be used to obtain the following critical stability constraints.

Theorem 5: Given (10), the largest vicinity V_{k_o} of stability of $D(z;k)$ obeys, and can be determined by, either of the two sets of stability constraints

i) $D_n(1;k)>0$, $(-1)D_n(-1;k)>0$, $c_{1,1}(k)>0$
ii) $c_{0,0}(k)>0$

where $c_{m,i}(k)$ are obtained for $D(z;k)$ by Algorithm 2.

Proof: The set i) corresponds to the critical constraints that Jury obtained in [1] by the aforementioned fact that $c_{1,1}$ is the determinant of the $(n-1)\times(n-1)$ leading submatrix of the SCF matrix [3]. A proof for ii) that is simultaneously also an alternative proof for i) can follow from recognizing Algorithm 2 as a greatest common divisor algorithm between $D_n(z)$ and $D_n^\sharp(z)$. Their common zeros are either UC or RP zeros. A total of s such zeros imply a symmetric $C_s(z)$ followed by a $C_{s-1}(z)\equiv 0$. Similar to the characterization of Algorithm 1 in Lemma 1, there is here too an "if and only if" relation between this situation and $D_n(z)$ having s UC or RP zeros. Since the algorithm is run with literal coefficients and is known to be stable $k=k_o$ normal termination of the algorithm is granted. Thus, condition i) is implied by an argument similar to the proof for Theorem 3. When $c_{1,1}(k)$ vanishes also $c_{0,0}(k)$ vanishes. $c_{0,0}(k)$ will vanish also for any zero on the unit-circle that $D_n(1;k)$ and $D_n(-1;k)$ are meant to detect in the set i). It follows that the single condition ii) is enough to maintain stability. ∎

Note that i) and ii) look like 1) and 3), respectively, in Theorem 3. The set 2) there is a special bonus that the immittance approach admits (due to the special role that $z=1$ plays in it.) Again, the single constraint i.e., ii) is less rewarding for determining stability constraints than the original set of constraints i) that Jury proposed.

Example: Suppose it is known that (6) is stable for some nominal value of its coefficients (say $K=1$). Then, according to Theorem 5, the stable vicinity of this nominal value can be determined by the examination of only three conditions: the two in (11) plus the constraint $c_{1,1}(K)>0$ on the polynomial of degree 12 in K in (14). The alternative (ii) suggests to examine instead just the single constraint $c_{0,0}(K)>0$, the last polynomial of degree 14 in the list of constraints (14).

There is less computation involved in obtaining the set of inequalities (7) than required to obtain the set (14) for a superposition of two reasons. One stems from the fact that Algorithm 1 runs symmetric polynomials and thus requires the calculation of only half of the coefficients. This is the usual advantage of immittance algorithms over corresponding scattering algorithms – an inherent ability to capture symmetry and exploit it to reduce the amount of computation. The second reason stems from the fact that Algorithm 1 creates constraints that are polynomials of lower degree in K, a direct consequence of the fact that it features a more restrained growth of coefficients size.

The fact that the growth rate of the coefficients in the MJT test is two times higher than their growth in Algorithm 1 implies that the constraints (that we remind that for k different literal coefficients imply k-variate polynomials) will appear with double degree than respective constraints obtained with the new BT. This has a more severe impact on the use of the MJT for stability constraints because it complicates a subsequent effort to extract from the constraints simplified expressions for the stable vicinity V_{k_o}.

IV. Conclusion

The paper considered critical stability constraints for discrete-time linear systems. It introduced a new approach that is based on an efficient IP variant of the BT. The integer-preservation property offers stability constraints for a polynomial with literal coefficients in the form of polynomials instead of rational functions when using a test without this property. Its efficiency is due partly to exploiting symmetry and mostly to featuring a linear growth of coefficients as function of the degree of the tested polynomial (as opposed to a potentially exponential growth in a naive integer preserving algorithm). The growth rate of coefficients is more restrained (by a factor of two) than with the MJT (that is too an efficient IP stability test). As a consequence, the proposed method outperforms the determination of critical constraints with the MJT not just by the familiar advantage of immittance algorithms over scattering algorithms (that stems from exploiting symmetry) but by also producing critical constrains that are polynomials of the literal coefficients with half of their degree in the MJT.

More research is needed to find whether the latter advantage indicates that there is a better scattering-type stability test yet to be discovered or it indicates that the immittance approach possesses some qualities that have no scattering counterpart.

References

[1] E. I. Jury, *Theory and Application of the Z-Transform Method*. New York: Wiley, 1964.

[2] ——, *Inners and the Stability of Linear Systems*. New York: Wiley, 1982.

[3] Y. Bistritz, "Reflection on Schur–Cohn matrices and Jury–Marden tables and classification of related unit-circle zero location criteria," *Circuits Syst. Signal Process.*, vol. 15, no. 1, pp. 111–136, 1996.

[4] ——, "Zero location with respect to the unit-circle of discrete-time linear system polynomials," *Proc. IEEE*, vol. 72, no. 9, pp. 1131–1142, Sep. 1984.

[5] ——, "Zero location of polynomials with respect to the unit-circle unhampered by nonessential singularities," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 3, pp. 305–314, Mar. 2002.

[6] ——, "An efficient integer-preserving stability test for discrete-time systems," *Circuits Syst. Signal Process.*, vol. 23, no. 3, pp. 195–213, 2004.

[7] E. I. Jury, "A modified stability table for linear discrete systems," *Proc. IEEE*, vol. 53, no. 1, pp. 184–185, Jan. 1965.

[8] ——, "A note on the modified stability table for linear discrete time system," *IEEE Trans. Circuits Syst.*, vol. 38, no. 2, pp. 221–223, Feb. 1991.

[9] ——, "The roles of Sylvester and Bezoutian matrices in the historical study of stability of linear discrete-time systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 12, pp. 1233–1251, Dec. 1998.

[10] P. G. Anderson, M. R. Garey, and L. E. Heindel, "Combinational aspects of deciding if all roots of a polynomial lie within the unit-circle," in *Computing*. New York: Springer-Verlag, 1976, vol. 16, pp. 293–304.

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

[12] ——, "Application of polynomial array method to discrete-time system stability," *Proc. Inst. Elect. Eng. D*, vol. 140, no. 3, pp. 198–204, May 1993.

[13] Y. Bistritz, "Real-polynomial based immittance-type tabular stability test for two-dimensional discrete systems," *Circuits Syst. Signal Process.*, vol. 22, no. 3, pp. 255–276, 2003.