Korean J. Chem. Eng., 32(9), 1733-1743 (2015)
DOI: 10.1007/s11814-014-0385-8

pISSN: 0256-1115
eISSN: 1975-7220

A novel and computationally efficient algorithm for stability analysis of multi input-multi output process control systems

Hooman Fatoorehchi*,, Hossein Abolghasemi*,**,†, Reza Zarghami**,***, Randolph Rach****, and Sebastian von Freeden*****

*Center for Separation Processes Modeling and Nano-Computations, School of Chemical Engineering, College of Engineering, University of Tehran, Tehran 11155-4563, Iran
**Advanced Process Control Research Laboratory, School of Chemical Engineering, College of Engineering, University of Tehran, Tehran 11155-4563, Iran
***Multiphase Systems Research Laboratory, School of Chemical Engineering, College of Engineering, University of Tehran, Tehran 11155-4563, Iran
****The George Adomian Center for Applied Mathematics, 316 South Maple Street, Hartford, Michigan 49057-1225, U.S.A.
*****Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW), University of Stuttgart, Stuttgart 70174, Germany
(*Received 26 September 2014 • accepted 22 December 2014*)

Abstract−An efficient method based on the Faddeev-Leverrier algorithm combined with the Adomian decomposition method is devised to facilitate the stability analysis of multi-input multi-output control systems. In contrast to prior eigenvalue algorithms, our method affords all eigenvalues of the state matrix, either real or complex. Specifically, the calculation of the complex eigenvalues is made possible through special canonical forms, mainly involving square root operators, of the characteristic equation of the state matrix. Moreover, the proposed method does not require an initial guess, which is often a matter of concern since an inappropriate guess can cause failure in such available schemes. For the sake of illustration, a number of numerical examples, including chemical reaction processes, are also provided that demonstrate the efficiency of our new technique.

Keywords: Control System, Stability Analysis, State-space Model, Adomian Decomposition Method, Adomian Polynomials

INTRODUCTION

Control systems are ubiquitous in chemical plants, where complex reactions are involved. Stability, on the other hand, is a basic requirement in the design of any control system. In fact, the stability of a control system is often critically important and is justifiably regarded as a safety issue. An example would be the control of a nuclear reactor, where an instability could cause a dreadful catastrophe. In general, the stability of a system is related to its response to external inputs or disturbances. Several well-known techniques for stability analysis of linear time invariant systems include the Routh-Hurwitz stability criterion, the root locus method, the Jury stability criterion, the Nyquist stability criterion and the Bistritz stability criterion [1-8].

The state-space representation of a linear dynamic system can provide insight into its stability status. This is possible by performing an eigenvalue analysis of the state matrix of the system under investigation. The problem of finding the eigenvalues of a specified matrix has been addressed, but not completely satisfactorily, for years. Several eigenvalue algorithms, namely the power method, the matrix squaring method, the Rayleigh quotient iteration method, and the Arnoldi iteration method, have appeared in the literature. Unfortunately, all of the aforementioned methods suffer from various drawbacks [9-12]. One of their principal demerits is their inability to compute complex eigenvalues of matrices. In addition, most of the available algorithms are incapable of affording all eigenvalues of a given matrix, besides depending upon the unreliable assumption of an initial guess.

We propose a new method to facilitate the stability analysis of dynamic control systems through the study of eigenvalues. Our new approach is based on the Faddeev-Leverrier algorithm combined with the Adomian decomposition method (ADM). In expansion of the concept first considered in [13], our proposed method is capable of efficiently computing complex eigenvalues and hence offers a complete stability analysis. For a better illustration, we present several examples in the sequel.

PRELIMINARIES

1. Theorem

A linear time invariant (LTI) multi input-multi output (MIMO) system characterized by the following state-space representation

$$\begin{cases} \dot{x} = \mathbf{A}x + \mathbf{B}u, \\ y = \mathbf{C}x + \mathbf{D}u, \end{cases} \tag{1}$$

is BIBO stable if all eigenvalues of the matrix $\mathbf{A}$, denoted by λ_i, are

†To whom correspondence should be addressed.
E-mail: abolghasemi.ha@gmail.com, hoab@ut.ac.ir

contained in the open left half-plane (LHP), i.e., $\mathrm{Re}(\lambda_i)<0\ \forall$ i. A matrix **A** with such a property is called stable or Hurwitz stable.

Proof. See [14].

2. The Faddeev-Leverrier Algorithm

Faddeev-Leverrier's algorithm is a powerful iterative tool for calculation of the characteristic polynomial of matrices [15,16]. Let Λ be an n-by-n matrix, then this algorithm consists of the following computational steps:

$$\begin{cases} \Lambda_1=\Lambda, \\ \Lambda_{i+1}=\Lambda(\Lambda_i+a_i\mathbf{I}), \ \text{for} \ 1\le i<n, \end{cases} \tag{2}$$

where

$$\begin{cases} a_0=1, \\ a_i=-\dfrac{\mathrm{trace}(\Lambda_i)}{i}, \ \text{for} \ 1\le i\le n. \end{cases} \tag{3}$$

In this way, all of the n+1 coefficients of $\phi(x)=\sum_{i=0}^{n}a_i x^{n-i}$, i.e., the characteristic polynomial of Λ, are found recursively through Eqs. (2) and (3).

As an additional benefit of the Faddeev-Leverrier algorithm, the inverse matrix Λ^{-1} can be obtained at no extra computational expense by $\Lambda^{-1}=-(1/an)(\Lambda_{n-1}+a_{n-1}\mathbf{I})$.

Before proceeding, we present a brief review of the basics of the ADM for the convenience of the reader.

3. Fundamentals of the ADM

Consider, without loss of generality, the following functional equation,

$$u-N(u)=f, \tag{4}$$

where N is a nonlinear operator on a Banach space E, the system input f is a specified element of E and we are seeking the system output $u\in E$, which satisfies Eq. (4). Assuming that Eq. (4) has a unique solution for every $f\in E$, then the ADM decomposes the solution u as an infinite series $u=\sum_{i=0}^{\infty}u_i$ and the nonlinearity as $N(u)=\sum_{i=0}^{\infty}A_i$, where the A_i are called the Adomian polynomials and are defined as [17,18]

$$A_i=A_i(u_0, u_1, \ldots, u_i)=\frac{1}{i!}\frac{d^i}{d\lambda^i}N\left(\sum_{k=0}^{\infty}u_k\lambda^k\right)\Bigg|_{\lambda=0}. \tag{5}$$

By selecting the initial solution component as $u_0=f$, the classic ADM uses the following Adomian recursion relation to generate components of the solution as

$$\begin{cases} u_0=f, \\ u_{i+1}=A_i, \ \text{for} \ i\ge 0. \end{cases} \tag{6}$$

The convergence and reliability of the ADM have been ascertained in prior research; e.g., see [19-23].

Previously, Fatoorehchi and Abolghasemi [24] developed a new improved algorithm to rapidly generate the Adomian polynomials of any desired analytic nonlinear operator. The algorithm primarily relies on string functions and symbolic programming. By setting the symbolic variable $NON=u_0+u_1+u_2+\cdots+u_n$ and a sufficiently large integer n, the following function in MATLAB can return the Adomian polynomial components of a nonlinear operator acting upon NON.

Program AdomPoly: An alternative MATLAB code for calculation of the Adomian polynomials

```
function sol=AdomPoly(expression,nth)
Ch=char(expand(expression));
s=strread(Ch, '%s', 'delimiter', '+');
for i=1:length(s)
t=strread(char(s(i)), '%s', 'delimiter', '*()expUlogsinh');
t=strrep(t,'^','*');
    if length(t)~=2
p=str2num(char(t));
sumindex=sum(p)-p(1);
    else
sumindex=str2num(char(t));
    end
list(i)=sumindex;
end
A='';
for j=1:length(list)
    if nth==list(j)
A=strcat(A,s(j),'+');
    end
end
N=length(char(A))-1;
F=strcat ('%',num2str(N),'c%n');
sol=sscanf(char(A),F);
```

Other techniques for calculation of the Adomian polynomials are available in the literature [25-29]. There is also an extensive literature concerning the theory and application of the ADM in various science and engineering fields; e.g., see [30-64].

OUR PROPOSED METHOD

Let the following equation,

$$x^n+c_{n-1}x^{n-1}+\cdots+c_2x^2+c_1x+c_0=0, \tag{7}$$

be the characteristic equation of an n-by-n matrix, namely **A**. Once written in Adomian's operator-theoretic notation, Eq. (7) becomes

$$Lx+Nx=, \tag{8}$$

where $L=c_1$, $Nx=x^n+c_{n-1}x^{n-1}+\cdots+c_2x^2$ and $g=-c_0$.

Provided that $c_1\neq 0$, we can choose $L^{-1}=(1/c_1)$, and hence

$$L^{-1}Lx=L^{-1}g-L^{-1}Nx. \tag{9}$$

Thus,

$$x=-\frac{1}{c_1}(c_0)-\frac{1}{c_1}(x^n+c_{n-1}x^{n-1}+\cdots+c_2x^2), \tag{10}$$

or

$$x=-\frac{c_0}{c_1}-\frac{1}{c_1}x^n-\frac{c_{n-1}}{c_1}x^{n-1}-\cdots-\frac{c_2}{c_1}x^2. \tag{11}$$

Now, we calculate the first real root of Eq. (7), or, in other words, one real eigenvalue of the matrix A as $\mu_1=\sum_{i=0}^{\infty}x_i$, or approximately as $\mu_1\approx\sum_{i=0}^{m}x_i$, where

$$\begin{cases} x_0=-\dfrac{c_0}{c_1}, \\ x_{i+1}=-\dfrac{1}{c_1}\Theta_{(n,i)}-\dfrac{c_{n-1}}{c_1}\Theta_{(n-1,i)}-\cdots-\dfrac{c_2}{c_1}\Theta_{(2,i)}, \ \text{for} \ i\ge 0. \end{cases} \tag{12}$$

where the $\Theta_{(n,i)}$, $\Theta_{(n-1,i)}$, …, $\Theta_{(2,i)}$ denote the Adomian polynomials decomposing the nonlinear terms x^n, x^{n-1}, …, x^2 in Eq. (11), respectively.

As discussed in [13,51], other real roots of Eq. (7), or real eigenvalues of the matrix **A**, can be found by successively eliminating the previously calculated roots through appropriate synthetic division followed by reapplying the ADM. For instance, the following equation

$$\begin{aligned}&\frac{x^n+c_{n-1}x^{n-1}+\cdots+c_2x^2+c_1x+c_0}{x-\mu_1}=0\\&\quad=x^{n-1}+b_{n-2}x^{n-2}+\cdots+b_2x^2+b_1x+b_0,\end{aligned} \tag{13}$$

can be investigated for the remaining real eigenvalues of the matrix **A**. This is usually referred to as polynomial deflation in mathematical terminology [65]; in this regard, see Appendix A.

Also, in the case of a diverging series generated by the recurrence relation (12), which may indicate the presence of at least two complex conjugate roots [35], the change of variable $z=x-\alpha$ may enable calculation of the real-valued roots. As a systematic way, we select α such that it is located in one of the Gershgorin disks of the matrix **A**. Note that **A** can be either the original state matrix of the problem, when we face a diverging sequence in search of the first eigenvalue, or the companion matrix of the deflated polynomial obtained from Eq. (13), while we are after other eigenvalues (second, third, and so on).

Definition: Let the complex-valued matrix $\mathbf{A}\in\mathbb{C}^{n\times n}$ have entries denoted by a_{ij}. Let $R_i=\sum_{j\neq i}^{n}|a_{ij}|$, for $i\in\{1, \ldots, n\}$, be the sum of absolute values of the non-diagonal entries in the *i*-th row of **A**. The closed disk $D(a_{ii}, R_i)$, which is centered at the point on the real axis, is called a Gershgorin of the matrix **A**.

The Gershgorin circle theorem guarantees that the eigenvalues of **A** lie in its Gershgorin disks. More theoretical foundations, theorems and their proofs, in case of diverging series yielded from recurrence (12) are presented in [51].

Complex eigenvalues of the matrix **A** can be calculated in a similar way via the ADM, but through the use of other equivalent canonical forms of Eq. (7), which involve even-root expressions and the choice of zero for the first solution component. Specifically, one canonical form of Eq. (10), presuming that $c_2\neq0$, is

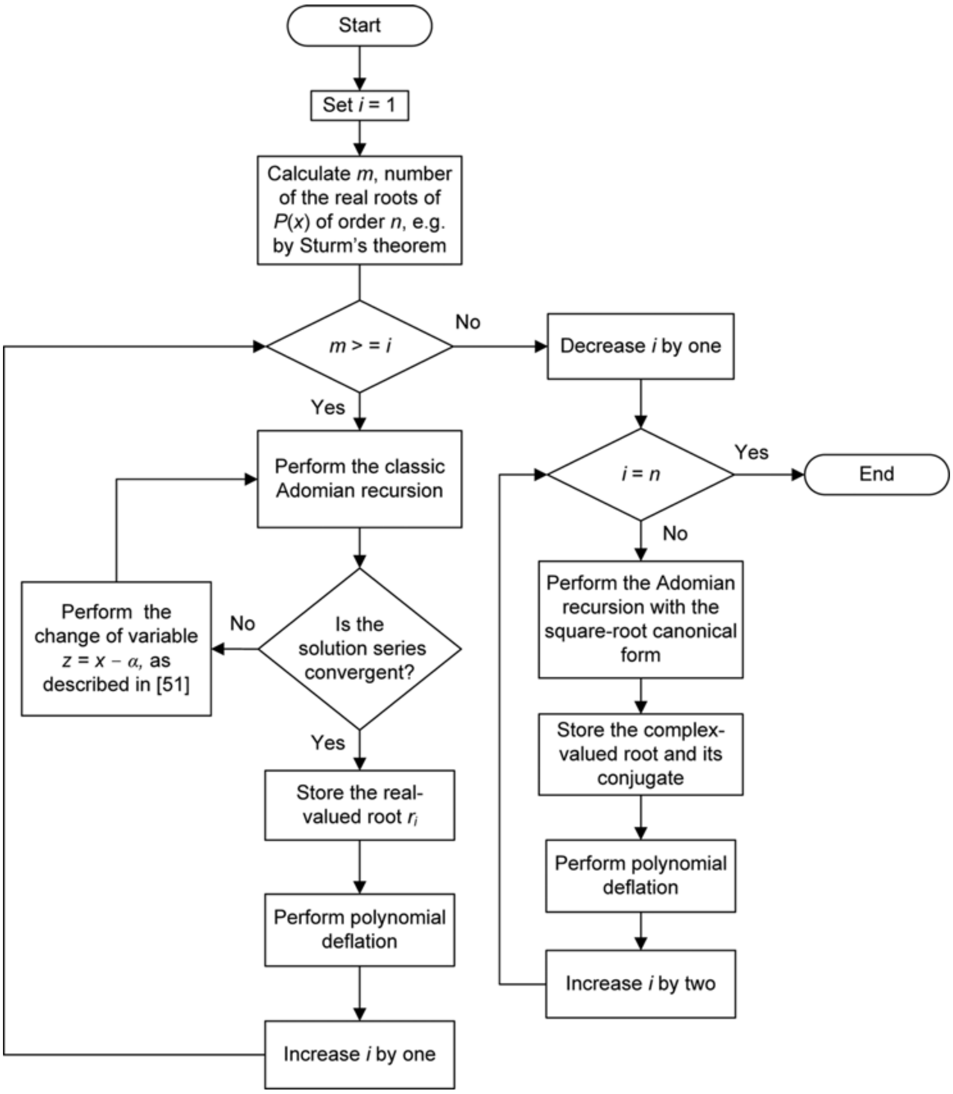

Fig. 1. Flowchart for the general algorithm of polynomial root-finding based on the ADM.

$$x=\sqrt{-\frac{1}{c_2}x^n-\frac{c_{n-1}}{c_2}x^{n-1}-\cdots-\frac{c_3}{c_2}x^3-\frac{c_1}{c_2}x-\frac{c_0}{c_2}}. \tag{14}$$

Consequently, the classic ADM generates the solution components by the recurrence relation as

$$\begin{cases} x_0=0, \\ x_{i+1}=\Phi_i,\ i\geq 0. \end{cases} \tag{15}$$

where the Φ_i are the Adomian polynomials, which decompose the square-root term on the right-hand side of Eq. (14). Accordingly, $\mu_2=\sum_{i=0}^{\infty}x_i$, or its truncated approximation $\tilde{\mu}_2=\sum_{i=0}^{m}x_i$, corresponds to one, if any, of the complex eigenvalues of the matrix **A**. Appendix B provides a more general formula for obtaining the complex roots of the polynomial equation P(z)=0 via its square-root type canonical form by the ADM. Finally, Fig. 1 depicts the flowchart for our new scheme, which is able to systematically calculate all the roots of a polynomial, either real or complex.

In view of the described procedure, it is clear that the sequence $(U_i=\sum_{j=0}^{i}x_j)_{i=0}^{N}$, where the x_j values are either obtained from Eq. (12) or Eq. (15), converges to an eigenvalue of the matrix **A** as N increases. Optionally, we can enhance the convergence rate of $(U_i)_{i=0}^{N}$ by employing a nonlinear sequence-to-sequence transformation proposed by Shanks [66]. The Shanks transform of the mentioned sequence is defined by

$$\mathrm{Sh}(U_i)=\frac{U_{i+1}U_{i-1}-U_i^2}{U_{i+1}-2U_i+U_{i-1}}. \tag{16}$$

Successive implementation of the Shanks transforms, i.e., $\mathrm{Sh}^2(U_i)=\mathrm{Sh}(\mathrm{Sh}(U_i))$, $\mathrm{Sh}^3(U_i)=\mathrm{Sh}(\mathrm{Sh}(\mathrm{Sh}(U_i)))$, etc., will lead to further speedup. The computational advantage of the Shanks transform technique, as a post-treatment, when combined with the ADM has been demonstrated previously; in this regard see e.g. [49,52,54].

ILLUSTRATIVE EXAMPLES

For the reader's convenience, we have listed the first few components of the Adomian polynomials, which appear in the following examples in Appendix C.

1. Example 1

Consider a linear time invariant system described by

$$\begin{cases} \dot{x}=\begin{bmatrix} 0.0650 & -3.1057 & 1.6992 \\ 0.2525 & -5.9760 & 3.2666 \\ 1.2884 & -8.0607 & 4.0740 \end{bmatrix}x+\begin{bmatrix} -1.1210 \\ 0 \\ 0.5 \end{bmatrix}u, \\ y=\begin{bmatrix} 0 \\ -1.2750 \\ 2 \end{bmatrix}x. \end{cases} \tag{17}$$

By virtue of Faddeev-Leverrier's algorithm, we obtain the characteristic equation of the state matrix as

$$P(\lambda)=\lambda^3+1.837\lambda^2+0.4557\lambda+0.1224=0. \tag{18}$$

The simplest canonical form of Eq. (18) is

$$\lambda=-\frac{0.1224}{0.4557}-\frac{1}{0.4557}\lambda^3-\frac{1.837}{0.4557}\lambda^2. \tag{19}$$

The ADM suggests the following recurrence relation for a solution to Eq. (18) as

$$\begin{cases} \lambda_0=-\dfrac{0.1224}{0.4557}, \\ \lambda_{i+1}=-\dfrac{1}{0.4557}A_i(\lambda_0,\ldots,\lambda_i)-\dfrac{1.837}{0.4557}B_i(\lambda_0,\ldots,\lambda_i),\ \text{for } i\geq 0, \end{cases} \tag{20}$$

where the A_i and B_i are the Adomian polynomials that decompose the nonlinear terms λ^3 and λ^2, respectively.

By the recurrence relation (20), we calculate the first several solution components as

$\lambda_0=-0.2686$, $\lambda_1=-0.2483$, $\lambda_2=-0.4198$, $\lambda_3=-0.8492$, $\lambda_4=-1.8737$, $\lambda_5=-4.3503$, $\lambda_6=-10.4408$, $\lambda_7=-25.6431$, $\lambda_8=-64.0497$, $\lambda_9=-162.0213$, …

which is a diverging series, again possibly indicating the existence of at least two complex conjugate roots.

To circumvent this problem, we introduce the change of variable $\lambda=z-2$ to obtain

$$P(z)=z^3-4.1630z^2+5.1077z-1.4410=0, \tag{21}$$

or, in canonical form,

$$z=\frac{1.4410}{5.1077}-\frac{1}{5.1077}z^3+\frac{4.1630}{5.1077}z^2. \tag{22}$$

Now, the ADM generates a convergent sequence for a solution of Eq. (21) as $z_0=0.2821$, $z_1=0.0605$, $z_2=0.0250$, $z_3=0.0127$, $z_4=0.0072$, $z_5=0.0043$, $z_6=0.0027$, $z_7=0.0018$, $z_8=0.0012$, $z_9=7.8493\times10^{-4}$.

Hence, the first eigenvalue of the state matrix is found to be $\mu_1\approx\sum_{i=0}^{9}z_i-2=-1.6018$.

In search for the other two remaining eigenvalues, we remove the root $\mu_1\approx-1.6018$ from the polynomial Eq. (18) by synthetic division as:

$$\begin{aligned} Q(\lambda)&=\frac{\lambda^3+1.837\lambda^2+0.4557\lambda+0.1224}{\lambda+1.6018} \\ &\approx 0.9990\lambda^2+0.2368\lambda+0.0764, \end{aligned} \tag{23}$$

Next, we construct a pair of canonical forms from the equation Q=(λ) as

$$\lambda=\pm\sqrt{-\frac{0.2368}{0.9990}\lambda-\frac{0.0764}{0.9990}}. \tag{24}$$

Through the canonical equation $\lambda=\sqrt{-0.2368/0.9990\lambda-0.0764/0.9990}$, it is straightforward to calculate the solution components as

$$\begin{cases} \lambda_0=0, \\ \lambda_{i+1}=C_i(\lambda_0,\ldots,\lambda_i),\ \text{for } i\geq 0, \end{cases} \tag{25}$$

where the C_i denotes the Adomian polynomials representing the radical term. Recurrence relation (25) yields the second eigenvalue of the state matrix as

$$\mu_2 \approx \sum_{i=0}^{9} \lambda_i = -0.1185 + 0.2499\mathrm{i}. \tag{26}$$

A decomposition solution to the other canonical equation, namely $\lambda = -\sqrt{-0.2368/0.9990\lambda - 0.0764/0.9990}$, is given by $\sum_{i=0}^{\infty}\lambda_i$ with

$$\begin{cases} \lambda_0 = 0, \\ \lambda_{i+1} = D_i(\lambda_0, \ldots, \lambda_i), \text{ for } i \ge 0, \end{cases} \tag{27}$$

where the D_i are the Adomian polynomials for the negative radical term. Hence,

$$\mu_3 \approx \sum_{i=0}^{9} \lambda_i = -0.1185 - 0.2499\mathrm{i}. \tag{28}$$

In view of the values of μ_1, μ_2 and μ_3, we deduce that system (17) is BIBO stable since $\mathrm{Re}(\mu_i)<0$ for i=1, 2, 3.

We remark that the well-known fixed-point iteration scheme fails to correctly calculate any of the complex roots, i.e., by Eq. (24), and returns grossly erroneous imaginary parts. For example, in this numerical example, the aforementioned method yields 0.1185−0.3444i, which is a misleading result for the designer.

2. Example 2. The Hydrolysis of Propylene Oxide in a CSTR

Bakošová et al. [67] provided an LTI model for the hydrolysis of propylene oxide to propylene glycol, according to the reaction $C_3H_6O \rightarrow C_3H_8O_2$, in a continuously stirred tank reactor.

The model has a state-space representation at the reaction temperature of T_r=343.1 K as follows [67]:

$$\begin{cases} \dot{x} = \begin{bmatrix} -0.0664 & 0 & -0.0001 & 0 \\ 0.0365 & -0.0299 & 0.0001 & 0 \\ 54.9420 & 0 & 0.1329 & 0.0138 \\ 0 & 0 & 0.0144 & -0.3297 \end{bmatrix} x \\ \quad + \begin{bmatrix} 0.0188 & 0 \\ -0.0188 & 0 \\ -18.3005 & 0 \\ 0 & -1.1978 \end{bmatrix} u, \\ y = [0 \ \ 0 \ \ 1 \ \ 0]x. \end{cases} \tag{29}$$

The characteristic equation of the system matrix is readily obtained by the Faddeev-Leverrier algorithm as

$$P(\lambda) = \lambda^4 + 0.2931\lambda^3 - 0.0176\lambda^2 - 0.0019\lambda - 0.3322 \times 10^{-4} = 0, \tag{30}$$

which is equivalent to the canonical form

$$\lambda = \frac{1}{0.0019}\lambda^4 + \frac{0.2931}{0.0019}\lambda^3 - \frac{0.0176}{0.0019}\lambda^2 - \frac{0.3322 \times 10^{-4}}{0.0019}. \tag{31}$$

Applying the ADM, we come up with the following recurrence relation

$$\begin{cases} \lambda_0 = -\dfrac{0.3322 \times 10^{-4}}{0.0019}, \\ \lambda_{i+1} = \dfrac{1}{0.0019} A_i(\lambda_0, \ldots, \lambda_i) + \dfrac{0.2931}{0.0019} B_i(\lambda_0, \ldots, \lambda_i) \\ \quad - \dfrac{0.0176}{0.0019} C_i(\lambda_0, \ldots, \lambda_i), \text{ for } i \ge 0, \end{cases} \tag{32}$$

where the A_i, B_i and C_i are the Adomian polynomials that decompose the quartic, cubic and quadratic nonlinearities in Eq. (31), respectively. By virtue of Eq. (32), we compute the solution components as λ_0=−0.0175, λ_1=−0.0036, λ_2=−0.0016, $\lambda_3=-9.5717\times10^{-4}$, $\lambda_4=-6.3388\times10^{-4}$, $\lambda_5=-4.5244\times10^{-4}$, $\lambda_6=-3.3945\times10^{-4}$, $\lambda_7=-2.6391\times10^{-4}$, $\lambda_8=-2.1072\times10^{-4}$, $\lambda_9=-1.7179\times10^{-4}$, $\lambda_{10}=-1.4239\times10^{-4}$, $\lambda_{11}=-1.1964\times10^{-4}$.

Hence, we approximate $\mu_1 \approx \sum_{i=0}^{11}\lambda_i = -0.0260$. According to the flowchart shown in Fig. 1, we perform a polynomial deflation to yield

$$\begin{aligned} Q(\lambda) &= \frac{\lambda^4 + 0.2931\lambda^3 - 0.0176\lambda^2 - 0.0019\lambda - 0.3322 \times 10^{-4}}{\lambda + 0.0260} \\ &\approx 1.9019\lambda^3 + 0.2436\lambda^2 - 0.0239\lambda - 0.0013 = 0, \end{aligned} \tag{33}$$

or equivalently,

$$\lambda = \frac{1.9019}{0.0239}\lambda^3 + \frac{0.2436}{0.0239}\lambda^2 - \frac{0.0013}{0.0239}. \tag{34}$$

Applying the ADM, it is easy to solve Eq. (34) for another eigenvalue as $\mu_2 \approx \sum_{i=0}^{11}\lambda_i = -0.0422$.

We have to perform a second polynomial deflation, see Fig. 1, in pursuit of the third eigenvalue of the state matrix. Therefore,

$$\begin{aligned} R(\lambda) &= \frac{1.9019\lambda^3 + 0.2436\lambda^2 - 0.0239\lambda - 0.0013}{\lambda + 0.0422} \\ &\approx 1.8947\lambda^2 + 0.1636\lambda - 0.0308 = 0, \end{aligned} \tag{35}$$

or, in the canonical form,

$$\lambda = -\frac{1.8947}{0.1636}\lambda^2 + \frac{0.0308}{0.1636}. \tag{36}$$

It is straightforward to verify that the ADM leads to a diverging sequence in the case of Eq. (36). To remedy this situation, a change of variable λ=z+0.5 is necessary as described in Section 3. Hence, Eq. (36) is converted to

$$z = -\frac{1.8947}{2.0583}z^2 - \frac{0.5247}{2.0583}. \tag{37}$$

Now, the classic ADM provides the following solution components as z_0=−0.2549, z_1=−0.0598, z_2=−0.0281, z_3=−0.0165, z_4=−0.0108, z_5=−0.0076, z_6=−0.0056, z_7=−0.0043, z_8=−0.0034, z_9=−0.0027, z_{10}=−0.0022, z_{11}=−0.0018.

Consequently, we approximate the third eigenvalue as μ_3»0.5 $+\sum_{i=0}^{11} z_i = 0.1024$. Since $\mathrm{Re}(\mu_3)>0$, we conclude that the system is unstable as a direct corollary of Theorem 2.1. This is in complete agreement with the results in [67].

3. Example 3

Analyze the stability of an LTI system, whose state matrix is

$$\mathbf{A} = \begin{bmatrix} -8.5295 & -5.6879 & -8.0877 & -3.2082 \\ 21.7054 & 14.6472 & 21.7202 & 8.7086 \\ -5.3731 & -3.7067 & -5.8512 & -2.1293 \\ -4.7866 & -3.0641 & -3.7794 & -2.6665 \end{bmatrix}. \tag{38}$$

We first compute the characteristic equation of the matrix **A** via the Faddeev-Leverrier algorithm as

$$P(\lambda)=\lambda^4+2.4\lambda^3+2.3528\lambda^2+0.9503\lambda+0.0738=0. \tag{39}$$

Writing Eq. (41) in its canonical form, we obtain

$$\lambda=-\frac{0.0738}{0.9503}-\frac{1}{0.9503}\lambda^4-\frac{2.4}{0.9503}\lambda^3-\frac{2.3528}{0.9503}\lambda^2. \tag{40}$$

According to the methodology of the ADM, the following recurrence relation provides the components of one root of the four possible roots of Eq. (39):

$$\begin{cases} \lambda_0=-\dfrac{0.0738}{0.9503}, \\ \lambda_{i+1}=-\dfrac{1}{0.9503}A_i(\lambda_0, \ldots, \lambda_i)-\dfrac{2.4}{0.9503}B_i(\lambda_0, \ldots, \lambda_i) \\ \qquad -\dfrac{2.3528}{0.9503}C_i(\lambda_0, \ldots, \lambda_i), \text{ for } i\geq 0, \end{cases} \tag{41}$$

where the A_i, B_i and C_i are the Adomian polynomials for the nonlinear terms λ^4, λ^3 and λ^2, respectively.

By the recurrence relation (41), we easily obtain the following converging solution sequence $\lambda_0=-0.0777$, $\lambda_1=-0.0138$, $\lambda_2=-0.0047$, $\lambda_3=-0.0020$, $\lambda_4=-9.1434\times10^{-4}$, $\lambda_5=-4.5276\times10^{-4}$, $\lambda_6=-2.3404\times10^{-4}$, $\lambda_7=-1.2480\times10^{-4}$, $\lambda_8=-6.8134\times10^{-5}$, $\lambda_9=-3.7891\times10^{-5}$, $\lambda_{10}=-2.1389\times10^{-5}$.

So, it yields $\mu_1\approx\sum_{i=0}^{10}\lambda_i=-0.1000$.

We need to eliminate the prior root from $P(\lambda)=0$ in order to enable the calculation of additional roots. So, we let

$$Q(\lambda)=\frac{P(\lambda)}{\lambda+0.1000}=\lambda^3+2.3\lambda^2+2.1225\lambda+0.7380. \tag{42}$$

To facilitate the solution of $Q(\lambda)=0$ by the ADM, we let $\lambda=z-1$ and thus obtain

$$Q(z)=z^3-0.7z^2+0.5225z-0.0845=0. \tag{43}$$

The simplest equivalent canonical form to Eq. (43) is

$$z=\frac{0.0845}{0.5225}-\frac{1}{0.5225}z^3+\frac{0.7}{0.5225}z^2. \tag{44}$$

In accordance with the principles of the ADM, we calculate the first several solution components as $z_0=0.1617$, $z_1=0.0269$, $z_2=0.0076$, $z_3=0.0025$, $z_4=8.2782\times10^{-4}$, $z_5=2.8101\times10^{-4}$, $z_6=9.4085\times10^{-5}$, $z_7=3.0442\times10^{-5}$, $z_8=9.2587\times10^{-6}$, $z_9=2.5096\times10^{-6}$.

Therefore, we find $\mu_2\approx\sum_{i=0}^{9}z_i-1=-0.8000$.

In pursuit of another eigenvalue for the state matrix, we remove the root $\mu_2\approx-0.8000$ by calculating

$$R(\lambda)=\frac{Q(\lambda)}{\lambda+0.8000}=\lambda^2+1.5\lambda+0.9225. \tag{45}$$

To solve $R(\lambda)=0$, we propose the following two canonical equations

$$\lambda=\pm\sqrt{-1.5\lambda-0.9225} \tag{46}$$

For the equation $\lambda=\sqrt{-1.5\lambda-0.9225}$, we can set the following recurrence relation

$$\begin{cases} \lambda_0=0, \\ \lambda_{i+1}=D_i(\lambda_0, \ldots, \lambda_i), \text{ for } i\geq 0, \end{cases} \tag{47}$$

where the D_i are the Adomian polynomials pertaining to the positive radical term $\sqrt{-1.5\lambda-0.9225}$.

Consequently, we calculate $\mu_3\approx\sum_{i=0}^{13}\lambda_i=-0.7500+0.6010i$.

In a similar manner, we seek the fourth eigenvalue of the state matrix through the following recurrence relation

$$\begin{cases} \lambda_0=0, \\ \lambda_{i+1}=E_i(\lambda_0, \ldots, \lambda_i), \text{ for } i\geq 0, \end{cases} \tag{48}$$

where the E_i designate the Adomian polynomials for the nonlinear term $-\sqrt{-1.5\lambda-0.9225}$.

We calculate $\mu_4\approx\sum_{i=0}^{13}\lambda_i=-0.7500+0.6010i$ by the recurrence relation (48).

Finally, since all of the four eigenvalues of the state matrix **A** have negative real parts, we promptly conclude that the system under investigation is BIBO stable.

4. Example 4

We next investigate the stability of a LTI system, whose state matrix is given by

$$\mathbf{A}=\begin{bmatrix} 27.3367 & 3.3851 & -2.2051 & 34.5790 \\ -11.4310 & -2.1338 & 0.8211 & -13.7734 \\ 5.7529 & 0.6751 & -2.3469 & 5.1798 \\ -19.9133 & -1.6767 & 2.6499 & -24.8560 \end{bmatrix}. \tag{49}$$

As the first step, we apply Faddeev-Leverrier's algorithm to provide the characteristic equation of the matrix **A** as

$$P(\lambda)=\lambda^4+2\lambda^3+17\lambda^2+0.75\lambda+1=0, \tag{50}$$

whose two possible canonical forms are

$$\lambda=\pm\sqrt{-\frac{1}{17}\lambda^4-\frac{2}{17}\lambda^3-\frac{0.75}{17}\lambda-\frac{1}{17}}. \tag{51}$$

Following the ADM procedure, similarly to our previous examples, we compute $\mu_1\approx\sum_{i=0}^{10}\lambda_i=-0.0188+0.2428i$ and $\mu_2\approx\sum_{i=0}^{10}\lambda_i=-0.0188-0.2428i$ from the preceding two canonical equations.

Removing the previous two roots from the characteristic polynomial by synthetic division yields

$$\begin{aligned} Q(\lambda)&=\frac{\lambda^4+2\lambda^3+17\lambda^2+0.75\lambda-1}{(-0.0188+0.2428i)(-0.0188+0.2428i)} \\ &\approx 1.0885\lambda^2+1.9559\lambda+16.8619=0. \end{aligned} \tag{52}$$

To proceed, we convert Eq. (52) to its two equivalent canonical forms as

$$\lambda=\pm\sqrt{-\frac{1.9559}{1.0885}\lambda-\frac{16.8619}{1.0885}}. \tag{53}$$

Once again, we invoke the ADM to calculate the remaining eigenvalues as $\mu_3\approx\sum_{i=0}^{10}\lambda_i=-0.8984+3.8319i$ and $\mu_4\approx-0.8984-3.8319i$.

We have shown that the state matrix **A** has four complex eigenvalues with negative real parts. Therefore, according to Theorem 2.1, the system under investigation is BIBO stable.

5. Example 5

Given the state matrix of a LTI system as

$$A=\begin{bmatrix} 12.1954 & 0.2474 & -1.9275 & -4.5238 \\ 29.7155 & 09292 & -5.2313 & -11.2493 \\ 35.6798 & 1.9861 & -6.9074 & -14.3038 \\ 20.0776 & -0.2431 & -3.0188 & -7.0183 \end{bmatrix}, \tag{54}$$

determine whether it is BIBO stable or not.

We compute the characteristic equation of the matrix **A** by the Faddeev-Leverrier algorithm as

$$P(\lambda)=\lambda^4+0.8011\lambda^3-6.2342\lambda^2+1.3810\lambda-0.7686=0. \tag{55}$$

Let us first search for complex eigenvalues of the state matrix, if any. For this purpose, we construct the two equivalent canonical radical forms of Eq. (55) as

$$\lambda=\pm\sqrt{\frac{1}{6.2342}\lambda^4+\frac{0.8011}{6.2342}\lambda^3+\frac{1.3810}{6.2342}\lambda-\frac{0.7686}{6.2342}}, \tag{56}$$

Choosing the one with the positive sign, the following recurrence relation gives the solution components of a complex root to Eq. (55) as

$$\begin{cases}\lambda_0=0,\\ \lambda_{i+1}=A_i(\lambda_0, \ldots, \lambda_i), \text{ for } i\geq 0,\end{cases} \tag{57}$$

where the A_i are the Adomian polynomials replacing the radical term.

By Eq. (57), we rapidly obtain $\mu_1\approx\sum_{i=0}^{10}\lambda_i=0.1020+0.3383i$. Since the real part of one eigenvalue of the state matrix is positive, we can deduce that at least one of the system poles is located in the right-half of the *s*-plane and hence the system is not stable.

6. Example 6. The Pottmann and Seborg CSTR Problem

Pottmann and Seborg [68] proposed a mathematical model for the analysis of a CSTR involving a single exothermic irreversible reaction, A→B. The reactor is equipped with an internal cooling coil. Fig. 2 illustrates a schematic layout of the CSTR. The dynamics of this nonlinear system, at its nominal conditions, can be modeled as

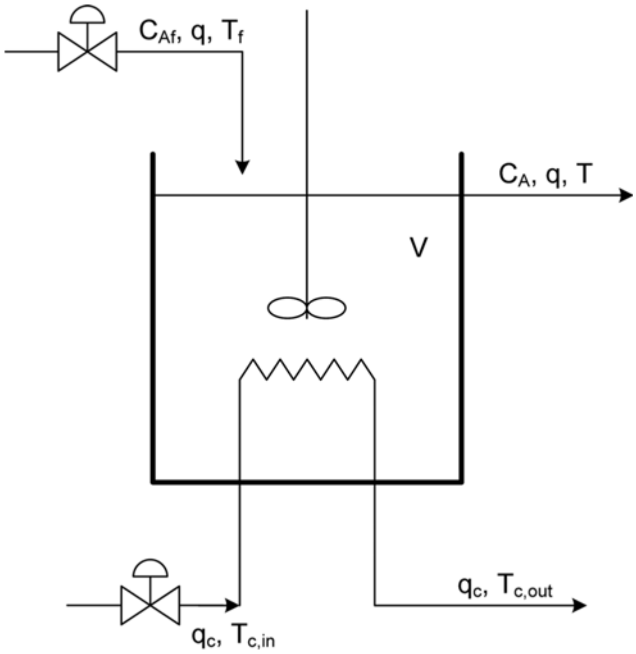

Fig. 2. The schematic layout of the chemical reactor system of Example 6.

$$\begin{cases}\dot{x}_1=\dfrac{u}{100}(1-x_1)-7.2\times10^{10}x_1\exp\left(-\dfrac{9.98\times10^3}{x_2}\right)\\ \dot{x}_2=\dfrac{u}{100}(350-x_2)+14.4\times10^{12}x_1\exp\left(-\dfrac{9.98\times10^3}{x_2}\right)\\ \quad+0.9991(350-x_2)\\ y=x_1\end{cases} \tag{58}$$

where x_1 denotes the effluent concentration of species A, x_2 is the reactor temperature and u is the feed flow rate [68]. With the aid of linearization, we obtain the state matrix of system (58) as follows:

$$\mathbf{A}=\begin{bmatrix}-\dfrac{u}{100}-7.2\times10^{10}\exp\left(-\dfrac{9.98\times10^3}{x_2}\right) & -7.1856\times10^{14}\dfrac{x_1}{x_2^2}\exp\left(-\dfrac{9.98\times10^3}{x_2}\right)\\ 1.44\times10^{13}\exp\left(-\dfrac{9.98\times10^3}{x_2}\right) & -\dfrac{u}{100}+1.43712\times10^{17}\dfrac{x_1}{x_2^2}\exp\left(-\dfrac{9.98\times10^3}{x_2}\right)-0.9991\end{bmatrix}. \tag{59}$$

Repeating our proposed procedure for any discretized point of the operating space $x_1\in[0, 1]$, $x_2\in[370, 450]$, $u\in[50, 150]$ by a simple code in MATLAB, we can determine the stability regions for the Pottmann and Seborg CSTR problem, similarly to Examples 1 to 5 in concept, as showed in Fig. 3.

ANALYSIS OF THE COMPUTATIONAL EFFICIENCY

In this section we compare the computational performance of our method with one of the most credited eigenvalue-finding algorithms known as the shifted QR algorithm [69,70]. The reason for choosing this specific algorithm is two-fold: 1) the shifted QR algorithm, like our proposed algorithm, can be applied to any matrices in general, and 2) the shifted QR algorithm, similar to our algorithm, furnishes all the real and complex eigenvalues of its input matrix. The shifted QR algorithm is briefly described as follows:

The Shifted QR Algorithm

I) Set the entry on the last row and last column of the matrix **A** as ω.
II) Decompose the matrix $\mathbf{A}-\omega\,\mathbf{I}$ as $\mathbf{A}-\omega\,\mathbf{I}=\mathbf{QR}$, where **R** is an upper triangular matrix (**I** is the identity matrix).
III) Substitute $\mathbf{A}=\mathbf{RQ}+\omega\,\mathbf{I}$.
IV) Repeat steps **I** to **II** until **A** converges to one of the following types:

$$\left[\begin{array}{c|c}\mathbf{S} & \mathbf{T}\\ \hline 0\;0\cdots0 & a\end{array}\right] \text{ or } \left[\begin{array}{c|c}\mathrm{U} & \mathrm{V}\\ \hline 0\;0\cdots0 & b\;c\\ 0\;0\cdots0 & d\;e\end{array}\right].$$

Either {keep a as an eigenvalue, substitute **A**=**S**}, or {solve the equation $\lambda^2-(b+e)\lambda+(be-cd)=0$ for λ to obtain two eigenvalues, then substitute **A**=**U**}. If **A** is two-by-two, then compute its eigenvalues according to the latter equation and terminate the algorithm otherwise go to step **I**.

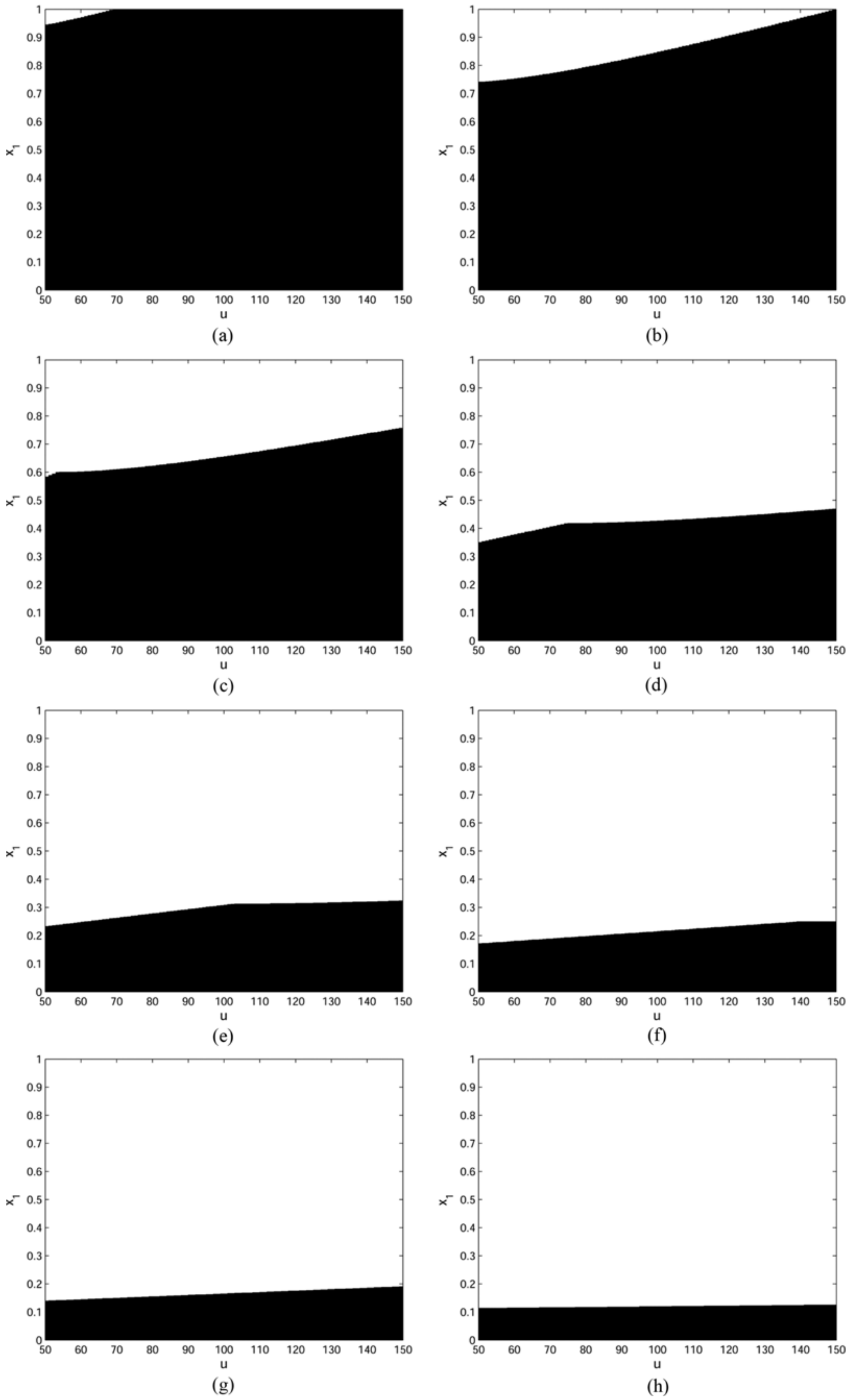

Fig. 3. Stability regions of the Pottmann and Seborg CSTR system at (a) x_2=370, (b) x_2=375, (c) x_2=380, (d) x_2=390, (e) x_2=400, (f) x_2=410, (g) x_2=420, and (h) x_2=450.

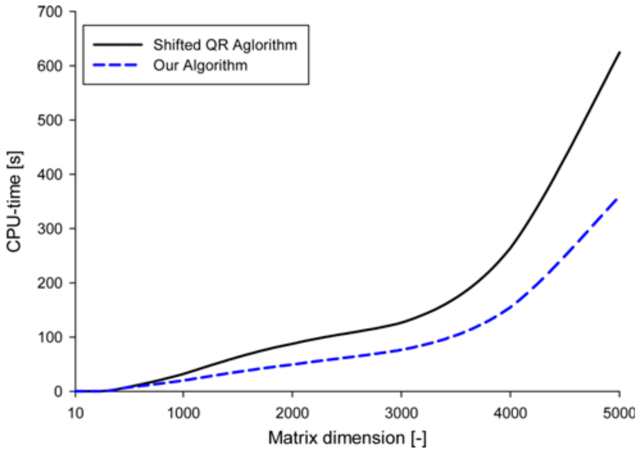

Fig. 4. The required CPU-times for calculation of all eigenvalues of randomly-generated matrices by the two algorithms. Iterations were terminated after a convergence tolerance of 10^{-10} was met.

Fig. 4 depicts the results of a comparative CPU-time analysis performed on randomly generated square matrices of different sizes using the MATLAB software package on a personal computer with a 2.66 GHz processor and 2 GB of RAM. As it can be seen in Fig. 4, our algorithm outperforms the shifted QR eigenvalue algorithm, particularly as the size of the input matrix enlarges, in terms of the required CPU-time. This can be justified by considering two facts: 1) The second step of the shifted QR algorithm entails an external QR decomposition algorithm, which adds up the computational burden of the Gram-Schmidt orthonormalization process. 2) There is no a priori knowledge about the computational speed of the fourth step of the shifted QR algorithm. In other words, this step involves a large number of matrix multiplications and tends to be remarkably time-consuming.

Finally, the idea of combining the Newton-Raphson algorithm with Faddeev-Leverrier's algorithm for the stability assessment of control systems is impractical because, in that way, the complex eigenvalues cannot be achieved. Moreover, the often disproportionate sensitivity of the Newton-Raphson algorithm to the initial guess values makes the hybrid approach unreliable and less robust.

CONCLUSION

An efficient scheme combining the Faddeev-Leverrier algorithm with the ADM was proposed to evaluate the stability of model-based control systems through an eigenvalue approach. Unlike the previous eigenvalue algorithms, our new method provides all eigenvalues of the state matrix, whether real or complex. In fact, the complex eigenvalue calculation ability is a novel and significant contribution of our paper in extending the technique described in [13]. Furthermore, as an additional advantage, our method obviates the need for an initial guess, which constitutes an intrinsic source of potential error in many previous solution strategies. Additionally, relatively simple programs can be coded to automate our new method to enable efficient, rapid calculation of state matrix eigenavlues, and hence expedite the stability analysis of large matrices representing state-space control models, where each requires a large number of variables.

ACKNOWLEDGMENT

We express our sincere gratitude to the Editor-in-Chief and anonymous reviewers of the Korean Journal of Chemical Engineering for their valuable comments and suggestions.

REFERENCES

1. A. K. Adebekun and F. J. Schork, *Chem. Eng. Commun.*, **69**, 43 (1988).
2. A. Arpornwichanop, P. Kittisupakorn and M. A. Hussain, *Korean J. Chem. Eng.*, **19**, 221 (2002).
3. D. R. Coughanowr and L. B. Koppel, *Process Systems Analysis and Control*, McGraw-Hill, New York (1965).
4. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Prentice-Hall, New Jersey (2001).
5. M. Gopal, *Digital Control Engineering*, John Wiley and Sons, New York (1988).
6. M. Han and D. E. Clough, *Korean J. Chem. Eng.*, **23**, 540 (2006).
7. K. Ogata, *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ (1999).
8. G. Stephanopoulos, *Chemical Process Control: An Introduction to Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ (1984).
9. G. A. Allaire and S. M. Kaber, *Numerical Linear Algebra*, Springer, New York (2002).
10. R. D. Pantazis and D. B. Szyld, *Numer. Lin. Algebra Appl.*, **2**, 251 (1995).
11. Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester (1992).
12. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford (1988).
13. H. Fatoorehchi and H. Abolghasemi, *J. Egyptian Math. Soc.*, **22**, 6 (2014).
14. M. Gopal, *Modern Control System Theory*, Wiley, New York (1993).
15. G. Helmberg and P. Wagner, *Lin. Algebra Appl.*, **185**, 219 (1993).
16. A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York (1964).
17. G. Adomian and R. Rach, *J. Math. Anal. Appl.*, **91**, 39 (1983).
18. G. Adomian, *Solving frontier problems of physics: The decomposition method*, Kluwer Academic, Dordrecht (1994).
19. K. Abbaoui and Y. Cherruault, *Math. Comput. Model.*, **20**, 69 (1994).
20. K. Abbaoui and Y. Cherruault, *Comput. Math. Appl.*, **28**, 103 (1994).
21. A. Abdelrazec and D. Pelinovsky, *Numer. Meth. Part. Differ. Equat.*, **27**, 749 (2011).
22. E. Babolian and J. Biazar, *Appl. Math. Comput.*, **130**, 383 (2002).
23. Y. Cherruault and G. Adomian, *Math. Comput. Model.*, **18**, 103 (1993).
24. H. Fatoorehchi and H. Abolghasemi, *J. Appl. Comput. Sci. Math.*, **5**, 85 (2011).
25. E. Babolian and Sh. Javadi, *Appl. Math. Comput.*, **153**, 253 (2004).
26. J.-S. Duan, *Appl. Math. Comput.*, **216**, 1235 (2010).
27. J.-S. Duan, *Appl. Math. Comput.*, **217**, 6337 (2011).
28. R. Rach, *J. Math. Anal. Appl.*, **102**, 415 (1984).
29. R. Rach, *Kybernetes*, **37**, 910 (2008).

30. S. Abbasbandy, *Appl. Math. Comput.*, **145**, 887 (2003).
31. S. Abbasbandy and M. T. Darvishi, *Appl. Math. Comput.*, **163**, 1265 (2005).
32. G. Adomian, *J. Math. Anal. Appl.*, **102**, 402 (1984).
33. G. Adomian, *Nonlinear Stochastic Operator Equations*, Academic Press, New York (1986).
34. G. Adomian, *Math. Comput. Model.*, **13**, 17 (1990).
35. G. Adomian and R. Rach, *J. Math. Anal. Appl.*, **105**, 141 (1985).
36. G. Adomian and R. Rach, *J. Math. Anal. Appl.*, **112**, 136 (1985).
37. G. Adomian and R. Rach, *J. Math. Anal. Appl.*, **113**, 504 (1986).
38. G. Adomian and R. Rach, *Kybernetes*, **15**, 33 (1986).
39. G. Adomian, R. Rach and D. Sarafyan, *J. Math. Anal. Appl.*, **111**, 423 (1985).
40. L. Bougoffa and R. C. Rach, *Kybernetes*, **42**, 641 (2013).
41. L. Bougoffa, R. C. Rach and A. Mennouni, *Appl. Math. Comput.*, **218**, 1785 (2011).
42. L. Bougoffa, A. Mennouni and R. C. Rach, *Appl. Math. Comput.*, **219**, 4423 (2013).
43. J.-S. Duan, R. Rach, D. Baleanu and A.-M. Wazwaz, *Commun. Fract. Calc.*, **3**, 73 (2012).
44. A. M. A. El-Sayed and M. Gaber, *Phys. Lett.*, **359**, 175 (2006).
45. H. Fatoorehchi and H. Abolghasemi, *Adv. Nat. Appl. Sci.*, **5**, 26 (2011).
46. H. Fatoorehchi and H. Abolghasemi, *Appl. Appl. Math.*, **7**, 717 (2012).
47. H. Fatoorehchi and H. Abolghasemi, *Intermetallics*, **32**, 35 (2013).
48. H. Fatoorehchi and H. Abolghasemi, *Appl. Math. Model.*, **37**, 6008 (2013).
49. H. Fatoorehchi and H. Abolghasemi, *J. Taiwan Inst. Chem. Eng.*, **45**, 880 (2014).
50. H. HFatoorehchi, R. Rach, O. Tavakoli and H. Abolghasemi, *Chem. Eng. Commun.*, **202**, 402 (2015).
51. H. Fatoorehchi and H. Abolghasemi, *J. Egyptian Math. Soc.*, **22**, 524 (2014).
52. H. Fatoorehchi, H. Abolghasemi, R. Rach and M. Assar, *Can. J. Chem. Eng.*, **92**, 2211 (2014).
53. H. Fatoorehchi, I. Gutman and H. Abolghasemi, *Kragujevac J. Sci.*, **36**, 69 (2014).
54. H. Fatoorehchi, H. Abolghasemi and R. Rach, *J. Petrol. Sci. Eng.*, **117**, 46 (2014).
55. R. Rach, J.-S. Duan and A.-M. Wazwaz, *J. Math. Chem.*, **52**, 255 (2014).
56. A. Dib, A. Haiahem and B. Bou-said, *Powder Technol.*, **269**, 193 (2015).
57. R. Rach, J.-S. Duan and A.-M. Wazwaz, *Chem. Eng. Commun.*, (In press), DOI:10.1080/00986445.2014.900054.
58. A. Hasseine, Z. Barhoum, M. Attarakih and H.-J. Bart, *Adv. Powder Technol.*, (In press), DOI:10.1016/j.apt.2014.08.011.
59. B. Kundu, *Int. J. Refrig.*, **32**, 1657 (2009).
60. B. Kundu and S. Wongwises, *J. Franklin Inst.*, **349**, 966 (2011).
61. R. Rach, *Kybernetes*, **41**, 1087 (2012).
62. A. Saravanan and N. Magesh, *J. Egyptian Math. Soc.*, **21**, 259 (2013).
63. A. M. Siddiqui, M. Hameed, B. M. Siddiqui and Q. K. Ghori, *Comm. Nonlinear Sci. Numer. Simulat.*, **15**, 2388 (2010).
64. A.-M. Wazwaz, R. Rach and J.-S. Duan, *Appl. Math. Comput.*, **219**, 5004 (2013).
65. J. D. Hoffman and S. Frankel, *Numerical Methods for Engineers and Scientists*, McGraw-Hill, New York (2001).
66. D. Shanks, *J. Math. Phys. Sci.*, **34**, 1 (1955).
67. M. Bakošová, D. Puna, P. Dostál and J. Závacká, *Chem. Paper Chem. Zvesti*, **63**, 527 (2009).
68. M. Pottmann and D. E. Seborg, *J. Proc. Cont.*, **2**, 189 (1992).
69. D. S. Watkins, *Fundamentals of Matrix Computations*, Wiley, New York (1991).
70. R. Bronson and G. B. Costa, *Linear algebra: An introduction*, Academic Press, New York (2007).

APPENDIX A. POLYNOMIAL DEFLATION BY SYNTHETIC DIVISION BASED ON THE FUNDAMENTAL THEOREM OF ALGEBRA

$$P_n(z)=\prod_{j=1}^{n}(z-r_j)=0,$$

$$P_{n-1}(z)=\frac{P_n(z)}{(z-r_1)}=\prod_{j=2}^{n}(z-r_j)=0,$$

$$\vdots$$

$$P_{n-k}(z)=\frac{P_{n-k+1}(z)}{(z-r_k)}=\frac{P_n(z)}{(z-r_1)\cdots(z-r_k)}=\frac{P_n(z)}{\prod_{j=1}^{k}(z-r_j)}$$

$$=\prod_{j=k+1}^{n}(z-r_j)=0,\quad \text{for}\quad 0\le k<n-1,$$

$$\vdots$$

$$P_2(z)=\frac{P_3(z)}{(z-r_{n-2})}=\frac{P_n(z)}{(z-r_1)\cdots(z-r_{n-2})}=\prod_{j=n-1}^{n}(z-r_j)=0,$$

$$P_1(z)=\frac{P_2(z)}{(z-r_{n-1})}=\frac{P_n(z)}{(z-r_1)\cdots(z-r_{n-1})}=\prod_{j=n}^{n}(z-r_j)=(z-r_n)=0$$

Thus, we can write

$$P_n(z)=\prod_{j=1}^{n}(z-r_j)=\sum_{j=0}^{n}c_{0,\,n-j}z^{n-j}=\sum_{j=0}^{n}c_{0,\,j}z^{j}=0,$$

$$P_{n-1}(z)=\prod_{j=2}^{n}(z-r_j)=\sum_{j=0}^{n-1}c_{1,\,n-1-j}z^{n-1-j}=\sum_{j=0}^{n-1}c_{1,\,j}z^{j}=0,$$

$$\vdots$$

$$P_{n-k}(z)=\prod_{j=k+1}^{n}(z-r_j)=\sum_{j=0}^{n-k}c_{k,\,n-k-j}z^{n-k-j}$$

$$=\sum_{j=0}^{n-k}c_{k,\,j}z^{j}=0,\quad \text{for}\quad 0\le k\le n-1,$$

$$\vdots$$

$$P_2(z)=\prod_{j=n-1}^{n}(z-r_j)=\sum_{j=0}^{2}c_{n-2,\,2-j}z^{2-j}=\sum_{j=0}^{2}c_{n-2,\,j}z^{j}=0,$$

$$P_1(z)=\prod_{j=n}^{n}(z-r_j)=\sum_{j=0}^{1}c_{n-1,\,1-j}z^{1-j}=\sum_{j=0}^{1}c_{n-1,\,j}z^{j}=(z-r_n)=0,$$

APPENDIX B. GENERAL FORMULA FOR CALCULATION OF THE COMPLEX ROOTS OF A POLYNOMIAL BY THE ADM

$$P_n(z)=0\Rightarrow z=\pm\sqrt{-\frac{c_{0,0}}{c_{0,2}}-\frac{c_{0,1}}{c_{0,2}}z-\sum_{j=3}^{n}\frac{c_{0,j}}{c_{0,2}}z^{j}}$$

$z=\sum_{n=0}^{\infty} z_n$, $z^j=\sum_{n=0}^{\infty} A_{j,n}$, $A_{j,n}=A_{j,n}(z_0, \ldots, z_n)$, and

$$\sqrt{-\frac{c_{0,0}}{c_{0,2}}-\frac{c_{0,1}}{c_{0,2}}z-\sum_{j=3}^{n}\frac{c_{0,j}}{c_{0,2}}z^j}=\sum_{n=0}^{\infty} A_{\frac{1}{2},n},\quad A_{\frac{1}{2},n}=A_{\frac{1}{2},n}(z_0, \ldots, z_n),$$

$$\sum_{n=0}^{\infty} z_n=\pm\sqrt{-\frac{c_{0,0}}{c_{0,2}}-\frac{c_{0,1}}{c_{0,2}}\sum_{n=0}^{\infty} z_n-\sum_{j=3}^{n}\frac{c_{0,j}}{c_{0,2}}\sum_{n=0}^{\infty} A_{j,n}}=\pm\sum_{n=0}^{\infty} A_{\frac{1}{2},n},$$

$$\begin{cases} z_0=0, \\ z_{n+1}=\pm A_{\frac{1}{2},n}, \quad n\geq 0. \end{cases}$$

Therefore,

$$z_1=A_{\frac{1}{2},0}=Z_{\frac{1}{2},0}(z_0)=\pm\sqrt{-\frac{c_{0,0}}{c_{0,2}}}=\pm i\sqrt{\frac{c_{0,0}}{c_{0,2}}},$$

since $z_0=0$ and $A_{j,n}=A_{j,n}(z_0)=z_0^j=0$, etc.

APPENDIX C. THE FIRST FIVE COMPONENTS OF THE ADOMIAN POLYNOMIALS FOR THE NONLINEARITIES APPEARED IN SECTION 4

Note that the calculations are done by Eq. (5).

$$N(u)=u^2=\sum_{i=0}^{\infty} A_i(u_0, \ldots, u_i)$$

$A_0(u_0)=u_0^2$, $A_1(u_0, u_1)=2u_0u_1$, $A_2(u_0, \ldots, u_2)=u_1^2+2u_0u_2$,
$A_3(u_0, \ldots, u_3)=2u_1u_2+2u_0u_3$, $A_4(u_0, \ldots, u_4)=u_2^2+2u_1u_3+2u_0u_4$.

$$N(u)=u^3=\sum_{i=0}^{\infty} A_i(u_0, \ldots, u_i)$$

$A_0(u_0)=u_0^3$, $A_1(u_0, u_1)=3u_0^2u_1$, $A_2(u_0, \ldots, u_2)=3u_0u_1^2+3u_0^2u_2$,
$A_3(u_0, \ldots, u_3)=u_1^3+6u_0u_1u_2+3u_0^2u_3$,
$A_4(u_0, \ldots, u_4)=3u_1^2u_2+3u_0u_2^2+6u_0u_1u_3+3u_0^2u_4$.

$$N(u)=u^4=\sum_{i=0}^{\infty} A_i(u_0, \ldots, u_i)$$

$A_0(u_0)=u_0^4$, $A_1(u_0, u_1)=4u_0^3u_1$, $A_2(u_0, \ldots, u_2)=6u_0^2u_1^2+4u_0^3u_2$,
$A_3(u_0, \ldots, u_3)=4u_0u_1^3+12u_0^2u_1u_2+4u_0^3u_3$,
$A_4(u_0, \ldots, u_4)=u_1^4+12u_0u_1^2u_2+6u_0^2u_2^2+12u_0^2u_1u_3+4u_0^3u_4$.

$$N(u)=\sqrt{au+b}=\sum_{i=0}^{\infty} A_i(u_0, \ldots, u_i)$$

$$A_0(u_0)=\sqrt{au_0+b},\quad A_1(u_0, u_1)=\frac{1}{2}\frac{au_1}{\sqrt{au_0+b}},$$

$$A_2(u_0, \ldots, u_2)=-\frac{1}{8}\frac{a^2u_1^2}{\sqrt{(au_0+b)^3}}+\frac{1}{2}\frac{au_2}{\sqrt{au_0+b}},$$

$$A_3(u_0, \ldots, u_3)=\frac{1}{16}\frac{a^3u_1^3}{\sqrt{(au_0+b)^5}}-\frac{1}{4}\frac{a^2u_1u_2}{\sqrt{(au_0+b)^3}}+\frac{1}{2}\frac{au_3}{\sqrt{au_0+b}},$$

$$A_4(u_0, \ldots, u_4)=-\frac{5}{128}\frac{a^4u_1^4}{\sqrt{(au_0+b)^7}}+\frac{3}{16}\frac{a^3u_1^2u_2}{\sqrt{(au_0+b)^5}}-\frac{1}{8}\frac{a^2u_2^2+2a^2u_1u_3}{\sqrt{(au_0+b)^3}}+\frac{1}{2}\frac{au_4}{\sqrt{(au_0+b)}}.$$

$$N(u)=\sqrt{au^4+bu^3+cu+d}=\sum_{i=0}^{\infty} A_i(u_0, \ldots, u_i)$$

$$A_0(u_0)=\sqrt{f},\quad A_1(u_0, u_1)=\frac{1}{2}\frac{g}{\sqrt{f}},$$

$$A_2(u_0, \ldots, u_2)=-\frac{1}{8}\frac{(g)^2}{\sqrt{f^3}}+\frac{1}{4}\frac{(12au_0^2u_1^2+8au_0^3u_2+6bu_0u_1^2+6bu_0^2u_2+2cu_2)}{\sqrt{f}}$$

$$A_3(u_0, \ldots, u_3)=\frac{1}{16}\frac{(g)^3}{\sqrt{f^5}}-\frac{1}{8}\frac{(g)(12au_0^2u_1^2+8au_0^3u_2+6bu_0u_1^2+6bu_0^2u_2+2cu_2)}{\sqrt{f^3}}+\frac{1}{12}\frac{(24au_0u_1^3+72au_0^2u_1u_2+24au_0^3u_3+6bu_1^3+36bu_0u_1u_2+18bu_0^2u_3+6cu_3)}{\sqrt{f}},$$

$$A_4(u_0, \ldots, u_4)=-\frac{5}{128}\frac{g^2}{\sqrt{f^7}}+\frac{3}{32}\frac{g^2(12au_0^2u_1^2+8au_0^3u_2+6bu_0u_1^2+6bu_0^2u_2+2cu_2)}{\sqrt{f^5}}-\frac{g}{24}\frac{(24au_0u_1^3+72au_0^2u_1u_2+24au_0^3u_3+6bu_1^3+36bu_0u_1u_2+18bu_0^2u_3+6cu_3)}{\sqrt{f^3}}+\frac{1}{48}\frac{\begin{pmatrix}24au_1^4+288au_0u_1^2u_2+144au_0^2u_2^2+288au_0^2u_1u_3+96au_0^3u_4\\+72bu_1^2u_2+72bu_0u_2^2+144bu_0u_1u_3+72bu_0^2u_4+24cu_4\end{pmatrix}}{\sqrt{f}},$$

where $f=au_0^4+bu_0^3+cu_0+d$, $g=4au_0^3u_1+3bu_0^2u_1+cu_1$.