POLYNOMIAL EVALUATIONS IN ENGINEERING

Khier Benmahammed¹, Saeed Badran² and Bassam Kourdi²

¹Department of electronics, LSI, University Ferhat Abbas Setif, 19000 Setif, Algeria.
²Electrical engineering department at Al Baha University, Al Baha, Saudia Arabia.

Senior Member, IEEE

ABSTRACT

Techniques for the evaluation of complex polynomials with one and two variables are introduced. Polynomials arise in many areas such as control systems, image and signal processing, coding theory, electrical networks, etc., and their evaluations are time consuming. This paper introduces new evaluationalgorithms that are straightforward with fewer arithmetic operations and a fast matrix exponentiation technique.

Keywords:
Complex polynomial, mapping, polynomial arithmetic, one or two variables, control systems, fast matrix exponentiation.

1. INTRODUCTION

Polynomials play an important role in almost all areas of engineering. Polynomials have wielded an enormous influence on the development of mathematics, since ancient times. Nowadays, polynomial models are ubiquitous and widely used across the sciences. They arise in robotics, coding theory, control systems, electrical networks, image and signal processing, mathematical biology, computer vision, game theory, statistics, and numerous other areas. In the last few decades, a tremendous improvements in microelectronics technology have led to an advancement in microprocessors which in their turn have increased the availability of low cost personal computers. The personal computers have had multiplicative effects on a number of areas such as control systems, signal processing, etc. However, at the beginning, many of the personal computers had compilers without the capability of complex arithmetic. Recent developments in computer software for computation have revolutionized the aforementioned fields. Formerly inaccessible problems are now tractable, providing fertile ground for simulations. This is a paper about the use of polynomials in engineering. We can do justice to only very small part of the subject and we confine most of our attention to the polynomial evaluation on a point of the complex plane. However, as the paper requires some basic notions of complex analysis, we have concentrated in the first two sections on building some theoretical foundations. Among which a
mapping from the complex variable domain to the skew symmetric matrix domain is studied and used in combination with the Horner technique to develop polynomial evaluation algorithms. These algorithms do not involve any complex arithmetic and require fewer floating point arithmetic operations than the conventional techniques. Furthermore, the algorithms can be used in the stability analysis of linear continuous time invariant systems. Following similar arguments as before, one can develop algorithms for the two variable polynomials. If the system is given in its state space representation, then using the Leverrier Faddeva algorithm, one can compute the transfer function. In this system framework, one often needs to calculate a matrix to a certain power. The latter operation is time consuming. Therefore, we have also provided a fast matrix exponentiation algorithm. The paper is structured as follows: the third section introduces the key concepts of complex variables needed in the forthcoming sections. The fourth section is devoted to the development of the real polynomial evaluation. The fifth section is focused on the generalization of the previously developed algorithm to include the complex polynomials. The sections six and seven are devoted to the arithmetics of a polynomial with its complex conjugate. The arithmetic operations, including the addition, subtraction, multiplication and division, and the differentiation of a polynomial evaluated at a given point on a complex plane, are often used in variety of areas, namely the adaptive robust optimal control and signal processing. The section eight shows how the algorithms introduced in the previous sections can be in control systems. The algorithms are used in the computation of the system frequency response. The latter is needed in order to draw the Bode plots or the Nyquist plot in the system stability analysis. In this section, a fast matrix exponentiation technique for a very large exponents is also introduced. Finally, section nine, devoted to the generalization of the previous techniques, presents an algorithm for the evaluation of 2 D polynomials.
2. NOTIONS OF COMPLEX VARIABLES

Let \( s = a + jb \) be a complex number belonging to the complex field \( \mathbb{C} \), with \( a \) as its real part and \( b \) its imaginary part and \( j = \sqrt{-1} \), [1].

Let

\[
R = \begin{bmatrix}
a & b \\
-b & a \\
\end{bmatrix}
\]  

(1)

be a rotation matrix belonging to the field of real skew symmetric matrices of rotation \( R \), with the properties:

- For \( s = a + jb \), \( R = \begin{bmatrix} \text{Re} \{s\} & \text{Im} \{s\} \\ -\text{Im} \{s\} & \text{Re} \{s\} \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \), where \( \text{Re} \{s\} \) is the real part and \( \text{Im} \{s\} \) is the imaginary part.

- \( R = aI + bJ \) where \( I \) is the identity matrix and \( J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \).

- \( RR^T = R^TR = (a^2 + b^2)I \). \( R \) is orthogonal.

- \( R^{-1} = \frac{1}{\sqrt{a^2 + b^2}} R^T \).

- Let \( S = \frac{1}{\sqrt{a^2 + b^2}} R \), then \( SJST = J \) and \( S \) is a symplectic matrix, [2].

The matrix \( J \) has the following properties:

- \( J^2 = -I \),
- \( J(-J) = I \),
- \( JJ^T = J(-J) = I \),
- \( JJ^T = J^TJ = I \).
- \( J^k = \begin{cases} 
-1 & \text{if } k = 4l + 2 = 2(1 + 2l), \\
-J & \text{if } k = 4l + 3 = 2(1 + 2l) + 1, \\
I & \text{if } k = 4l + 4 = 4(1 + l), \\
J & \text{if } k = 4l + 5 = 4(1 + l) + 1, 
\end{cases} \)

for \( l = 0, 1, 2, 3, \ldots \).
Notice that $R$ can be represented

$$R = \lambda_1 E_1 + \lambda_2 E_2,$$

(2)

where $\lambda_1 = a + jb$ and $\lambda_2 = \lambda_1^* = a - jb$ are the eigenvalues of $R$, and

$$E_1 = \frac{1}{2} \begin{bmatrix} 1 & -j \\ j & 1 \end{bmatrix} = \frac{1}{2} (I - jJ)$$

and

$$E_2 = E_1^* = \frac{1}{2} \begin{bmatrix} 1 & j \\ -j & 1 \end{bmatrix} = \frac{1}{2} (I + jJ),$$

are the spectral decomposition or skew Hermitian matrices. Therefore, for any known function $f(.)$, one has

$$f(R) = f(\lambda_1)E_1 + f(\lambda_2)E_2,$$

(3)

and of particular interest

$$R^k = \lambda_1^k E_1 + \lambda_2^k E_2,$$

(4)

$$= \sum_{m=0}^{k} \binom{k}{m} a^m b^{k-m} J^{k-m}.$$

Assume that $b \neq 0$ and let $T$ be a similarity transformation matrix

$$T = \begin{bmatrix} -a & 1 \\ -b & 0 \end{bmatrix},$$

such that

$$J_c = T^{-1}JT = \frac{1}{b} \begin{bmatrix} -a & 1 \\ -(a^2 + b^2) & a \end{bmatrix},$$

(6)

and

$$R_c = T^{-1}RT = \begin{bmatrix} 0 & 1 \\ -(a^2 + b^2) & 2a \end{bmatrix}.$$

(7)

Thus $J_c^k = T^{-1}J_c^kT$ and $R_c^k = T^{-1}R_c^kT$, for some integer $k$, where

$$J_c^k = \begin{cases} -I & \text{if } k = 4l + 2 = 2(1 + 2l), \\ -J_c & \text{if } k = 4l + 3 = 2(1 + 2l) + 1, \\ I & \text{if } k = 4l + 4 = 4(1 + l), \\ J_c & \text{if } k = 4l + 5 = 4(1 + l) + 1. \end{cases}$$

(8)

Thus, there is a mapping between the complex numbers $s$ and rotation matrices $R$,

$$s \longleftrightarrow R.$$

Such a map is one to one and onto, and therefore an isomorphism [2]. If $s_1 = a_1 + jb_1$ and $s_2 = a_2 + jb_2$ are two complex numbers in $C$, then $R_1 = a_1 I + b_1 J$ and $R_2 = a_2 I + b_2 J$ are in $R$, with the properties:

- $s_1 + s_2 = (a_1 + a_2) + j(b_1 + b_2) \in C$, then
  $$R_1 + R_2 = (a_1 + a_2)I + (b_1 + b_2)J = Re\{s_1 + s_2\}I + Im\{s_1 + s_2\}J \in R,$$

- $s_1 - s_2 = (a_1 - a_2) + j(b_1 - b_2) \in C$, then
  $$R_1 - R_2 = (a_1 - a_2)I + (b_1 - b_2)J = Re\{s_1 - s_2\}I + Im\{s_1 - s_2\}J \in R,$$

- $s_1 \ast s_2 = (a_1 a_2 - b_1 b_2) + j(a_2 b_1 + a_1 b_2) \in C$, then
  $$R_1 R_2 = (a_1 a_2 - b_1 b_2)I + (a_2 b_1 + a_1 b_2)J = Re\{s_1 \ast s_2\}I + Im\{s_1 \ast s_2\}J \in R,$$

- $\frac{s_1}{s_2} = \frac{a_1 a_2 + b_1 b_2}{a_2 a_2 + b_2 b_2} + j\frac{-a_1 b_2 + a_2 b_1}{a_2 a_2 + b_2 b_2} \in C$, then
  $$R_1 R_2^{-1} = R_2^{-1} R_1 = \frac{1}{a_2 a_2 + b_2 b_2} [(a_1 a_2 + b_1 b_2)I + (a_2 b_1 - a_1 b_2)J] = Re\{\frac{a_1}{a_2}\}I + Im\{\frac{b_1}{a_2}\}J \in R,$$

for $a_2 \neq 0$ and/or $b_2 \neq 0$.

In the next section, an algorithm for evaluating a polynomial in a given point will be developed.
3. POLYNOMIALS WITH REAL COEFFICIENTS

Suppose now that \( p(s) \) is an \( n^{th} \) order polynomial with real coefficients, of the form

\[
p(s) = \sum_{i=0}^{n} \alpha_i s^{n-i} - u_0 + jv_0.
\]  \hspace{1cm} (9)

Let \( w_0 = 0 \) such that \( p(s) = w_0 s^{n+1} + \sum_{i=0}^{n} \alpha_i w^{n-i} \) which can be written as

\[
p(s) = \alpha_n + s(\alpha_{n-1} + \cdots + s(\alpha_2 + s(\alpha_1 + s(\alpha_0 + sw_1)) + \cdots)),
\]  \hspace{1cm} (10)

\[w_1\]
\[w_2\]
\[\vdots\]
\[w_n\]
\[w_{n+1}\]

using Horner’s algorithm, one gets

\[
w_1 = sw_0 + \alpha_0
\]
\[
w_2 = sw_1 + \alpha_1
\]
\[
\vdots
\]
\[
w_n = sw_{n-1} + \alpha_{n-1}
\]
\[
w_{n+1} = sw_n + \alpha_n = p(s).
\]

In general, \( w_{l+1} = sw_l + \alpha_l \) for \( 0 \leq l \leq n \), with \( w_0 = 0 \) and \( w_{n+1} = p(s) \).

Since the mapping \( s \rightarrow R \) is an isomorphism, and

\[
s^l \leftrightarrow R^l = \begin{bmatrix} \text{Re} \{s^l\} & \text{Im} \{s^l\} \\ -\text{Im} \{s^l\} & \text{Re} \{s^l\} \end{bmatrix} = \text{Re} \{s^l\} I + \text{Im} \{s^l\} J
\]  \hspace{1cm} (11)

then one has

\[
p(s) \leftrightarrow p(R) = \begin{bmatrix} \text{Re} \{p(s)\} & \text{Im} \{p(s)\} \\ -\text{Im} \{p(s)\} & \text{Re} \{p(s)\} \end{bmatrix}
\]  

\[
= \text{Re} \{p(s)\} I + \text{Im} \{p(s)\} J
\]  

\[
= \sum_{i=0}^{n} \alpha_i R^{n-i}.
\]  \hspace{1cm} (12)

The real and imaginary parts of the polynomial \( p(s) \) can be determined by the following vector multiplication

\[
p(R) \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} \text{Re} \{p(s)\} \\ -\text{Im} \{p(s)\} \end{bmatrix}.
\]  \hspace{1cm} (13)
Let the vector \( w = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), since \( p(R_e) = T^{-1}p(R)T \) or
\[
Tp(R_e) = p(R)T,
\]
then the multiplication of both sides of the equation (14) by the vector \( w \) yields \( Tp(R_e)w = p(R)Tw \), but
\[
Tw = \begin{bmatrix} -a \\ -b \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.
\]
Therefore,
\[
p(R) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \text{Re} \{p(s)\} \\ -\text{Im} \{p(s)\} \end{bmatrix} = Tp(R_e)w.
\]
(15)

To determine the main equations of a polynomial evaluation procedure at a particular point, let \( z_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \), then the polynomial matrix equation
\[
p(R_c) = \sum_{i=0}^{n} \alpha_i R_c^{n-i},
\]
(17)
can be rewritten as
\[
p(R_c)w = R_c^{n+1}z_0 + \sum_{i=0}^{n} \alpha_i R_c^{n-i}w.
\]
(18)
Using Horner’s rules, but this time with a matrix as a variable, one gets
\[
z_{i+1} = R_c z_i + \alpha_i w, \quad \text{for } 0 \leq i \leq n.
\]
(19)
Hence, \( p(R_c)w = z_{n+1} \) and
\[
\begin{bmatrix} \text{Re} \{p(s)\} \\ -\text{Im} \{p(s)\} \end{bmatrix} = Tp(R_c)w = Tz_{n+1}.
\]
(20)

In the next section, an algorithm for evaluating in a given point, a polynomial with complex coefficients, will be developed.

4. POLYNOMIALS WITH COMPLEX COEFFICIENTS

Suppose that the \( n^{th} \) order polynomial \( p(s) \) has now complex coefficients, it can then be written as follows
\[
p(s) = \sum_{i=0}^{n} \gamma_i s^{n-i} = u_p + jv_p
\]
\[
= \sum_{i=0}^{n} (\alpha_i + j\beta_i) s^{n-i} = \sum_{i=0}^{n} \alpha_i s^{n-i} + j \sum_{i=0}^{n} \beta_i s^{n-i}
\]
\[
= p_\alpha(s) + j p_\beta(s) = (u_\alpha + jv_\alpha) + j(u_\beta + jv_\beta)
\]
\[
= (u_\alpha - v_\beta) + j(u_\alpha + u_\beta).
\]
(21)

From the mapping of \( p(s) = p_\alpha(s) + j p_\beta(s) \) onto \( p(R) = p_\alpha(R) + j p_\beta(R) \), one can get as it is done previously, the real and imaginary parts of \( p_\alpha(s) \) and \( p_\beta(s) \) as follows
\[
p_\alpha(R) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \text{Re} \{p_\alpha(s)\} \\ -\text{Im} \{p_\alpha(s)\} \end{bmatrix} = \begin{bmatrix} u_\alpha \\ -u_\alpha \end{bmatrix}
\]
(23)
\[
p_\beta(R) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \text{Re} \{p_\beta(s)\} \\ -\text{Im} \{p_\beta(s)\} \end{bmatrix} = \begin{bmatrix} u_\beta \\ -u_\beta \end{bmatrix}
\]
(24)
Since $R_e = T^{-1}RT$, then $p_a(R_e)T = Tp_a(R)T = Tp^a(R)$ and $p(R_e)T = Tp(R)$. We now define $U = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ and the exchange matrix $G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ such that $GUG = -U$, $GGU = U$, $UGU = -G$, $UUG = G$, $UG = J$ and $GU = -J$. However,

$$p(R) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} u_p \\ -v_p \end{bmatrix} = Tp(R_e)w$$ (25)

and

$$Tp(R_e)w = Tp_a(R_e)w + UGTp_3(R_e)w$$
$$= Tp_a(R_e)w + JTp_3(R_e)w,$$ (26)

where $Tp_a(R_e)w = \begin{bmatrix} u_\alpha \\ -u_\alpha \end{bmatrix}$ and $Tp_3(R_e)w = \begin{bmatrix} u_\beta \\ -v_\beta \end{bmatrix}$. To determine the main equations of the evaluation procedure of a polynomial with complex coefficients [4], let $z_0 = z_\beta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, then the polynomial matrix equations $p_a(R_e) = \sum_{l=0}^{n} \alpha_l R_e^{n-1}$ and $p_3(R_e) = \sum_{l=0}^{n} \beta_l R_e^{n-1}$ can be rewritten as

$$p_a(R_e)w = R_e^{n+1} z_\alpha + \sum_{l=0}^{n} \alpha_l R_e^{n-1}w$$ (27)

and

$$p_3(R_e)w = R_e^{n+1} z_\beta + \sum_{l=0}^{n} \beta_l R_e^{n-1}w.$$ (28)

Using again Horner's rules, one can have

$$z_{\alpha_{l+1}} = R_e z_\alpha + \alpha_l w, \quad \text{for} \quad 0 \leq l \leq n.$$ (29)

Similarly,

$$z_{\beta_{l+1}} = R_e z_\beta + \beta_l w, \quad \text{for} \quad 0 \leq l \leq n.$$ (30)

Hence, $p_a(R_e)w = z_{\alpha_{n+1}}$, $p_3(R_e)w = z_{\beta_{n+1}}$,

$$\begin{bmatrix} u_\alpha \\ -v_\alpha \end{bmatrix} = Tp_a(R_e)w = Tz_{\alpha_{n+1}}$$ (31)

and

$$\begin{bmatrix} u_\beta \\ -v_\beta \end{bmatrix} = Tp_3(R_e)w = Tz_{\beta_{n+1}}.$$ (32)

Therefore the real and imaginary parts of $p(s)$ can now be found as

$$\begin{bmatrix} u_p \\ v_p \end{bmatrix} = U T z_{\alpha_{n+1}} + G T z_{\beta_{n+1}}.$$ (33)

In the next section, the previously developed algorithm will be used to evaluate the sum, the subtraction, the product and the quotient of a complex polynomial with its complex conjugate in a given point.
5. ARITHMETIC OF COMPLEX POLYNOMIALS

Let the matrix \( V = \begin{bmatrix} 1 & -j \\ j & 1 \end{bmatrix} \) and the complex conjugate polynomial \( p^*(s^*) = u_p - jv_p \), then one can have the following polynomial vector

\[
\begin{bmatrix}
 p(s) \\
 p^*(s^*)
\end{bmatrix} = V \begin{bmatrix} Tz_{a_n+1} + JTz_{b_n+1} \end{bmatrix}.
\]

(34)

Since the absolute of a complex polynomial is given as

\[
|p(s)| = \sqrt{p(s)p^*(s^*)}
\]

(35)

and

\[
p(s)p^*(s^*) = |p(s)|^2,
\]

(36)

then

\[
p(s)p^*(s^*) = \frac{1}{2} \left( Tz_{a_n+1} + JTz_{b_n+1} \right)^T V^* G V \left( Tz_{a_n+1} + JTz_{b_n+1} \right)
\]

(37)

We now define the following matrices needed in the algebraic manipulation of the computational algorithm,

\[
A = \begin{bmatrix} a^2 + b^2 & -a \\ -a & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & b \\ -b & 0 \end{bmatrix},
\]

thus the product of \( p(s) \) by its complex conjugate is given as

\[
p(s)p^*(s^*) = |p(s)|^2,
\]

(38)

Using the previous results, one can easily find the reciprocal of \( p(s) \) as

\[
\frac{1}{p(s)} = \frac{p^*(s^*)}{|p(s)|^2}.
\]

(39)

Furthermore, the sum and the difference of \( p(s) \) with its complex conjugate are given as

\[
p(s) + p^*(s^*) = 2 \begin{bmatrix} 0 & 1 \end{bmatrix} \left( Az_{a_n+1} + Bz_{b_n+1} \right)
\]

(40)

and

\[
p(s) - p^*(s^*) = j2 \begin{bmatrix} 0 & 1 \end{bmatrix} \left( -Bz_{a_n+1} + Az_{b_n+1} \right),
\]

(41)

respectively.

6. DERIVATIVE OF A POLYNOMIAL

Since the \( n^{th} \) order polynomial \( p(s) \) is an entire function, it is analytic everywhere, then its derivative exists and can be computed using similar algorithms as the previous ones.

Note that \( p(s) = \sum_{l=0}^{n} a_l s^{n-l} = u_p + jv_p \), then

\[
\frac{dp(s)}{ds} = \sum_{l=0}^{n-1} a_l (n-l) s^{n-l-1},
\]

\[
= \sum_{l=0}^{n-1} p_l s^{n-l-1}
\]

(42)

\[
= \frac{du_p(s)}{ds} + j \frac{dv_p(s)}{ds}.
\]
Where \( \tau_l = (n - l) \alpha_l \).
Let \( \tilde{p}_r(s) = \frac{dp_r(s)}{ds} \), \( u_p(s) = \frac{du_p(s)}{ds} \) and \( v_p(s) = \frac{dv_p(s)}{ds} \), thus, if \( \tilde{p}_r(s) \) has
- real coefficients, then
\[
\begin{bmatrix}
u_p \\ v_p
\end{bmatrix} = UTz_{\alpha n},
\]
where in this case
\[
z_{\alpha l+1} = R_\alpha z_{\alpha l} + (n - l) \alpha_l w, \quad \text{for} \quad 0 \leq l \leq n - 1.
\] (44)
- complex coefficients
\[
\begin{bmatrix}
u_p \\ v_p
\end{bmatrix} = UTz_{\alpha n} + GTz_{\beta n+1},
\]
where in addition to \( z_{\alpha n} \) computed above, we also have
\[
z_{\beta l+1} = R_\beta z_{\beta l} + (n - l) \beta_l w, \quad \text{for} \quad 0 \leq l \leq n.
\] (46)
In the next section, the developed algorithms will be used in the computation of the control system frequency response needed in the stability tests.

7. APPLICATIONS AND MATRIX EXPONENTIATION

Assume that \( s = j\omega \) or \( \alpha = 0 \) and \( \beta = \omega \), and consider the following two cases:
1) the polynomial \( p(s) \) has complex coefficients, then its real and imaginary parts are obtained as
\[
\begin{bmatrix}
u_p \\ v_p
\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega & 0 \end{bmatrix} z_{\alpha l+1} + \begin{bmatrix} -\omega & 0 \\ 0 & 1 \end{bmatrix} z_{\beta l+1},
\] (47)
where
\[
z_{\alpha l+1} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} z_{\alpha l} + \alpha_l w
\] (48)
and
\[
z_{\beta l+1} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} z_{\beta l} + \beta_l w
\] (49)
for \( 0 \leq l \leq n \). Therefore the product and sum of \( p(j\omega) \) with its complex conjugate are determined as follows
\[
p(j\omega) + \overline{p(j\omega)} = |p(j\omega)|^2
\]
\[
- \begin{bmatrix} \tilde{T} \\ \tilde{T}_{\alpha l+1} \\ \tilde{T}_{\beta l+1}
\end{bmatrix} \begin{bmatrix} \omega^2 & 0 & 0 \\ 0 & 1 & -\omega \\ 0 & -\omega & \omega^2 \\ \omega & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{\alpha l+1} \\ z_{\beta l+1} \\ z_{\beta l+1} \end{bmatrix},
\] (50)
and
\[ p(j\omega) + p^*(-j\omega) = 2 \left( \begin{bmatrix} 0 & 1 \\ \omega & 0 \end{bmatrix} z_{L_{m+1}} + \begin{bmatrix} -\omega & 1 \\ 1 & -\omega \end{bmatrix} z_{R_{m+1}} \right). \] (51)

2) The polynomial \( p(s) \) has real coefficients, then \( z_{L_{1}} = 0 \) for \( 0 \leq l \leq n \), and its real and imaginary parts are determined as
\[ \begin{bmatrix} u_p \\ v_p \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega & 0 \end{bmatrix} z_{L_{m+1}} \] (52)

where
\[ z_{L_{m+1}} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} z_{L_{m+1}} + \alpha_{w} w. \] (53)

\[ p(j\omega)p^*(-j\omega) = z_{L_{m+1}}^T \begin{bmatrix} \omega^2 & 0 \\ 0 & 1 \end{bmatrix} z_{L_{m+1}} \] (54)

and
\[ p(j\omega) + p^*(-j\omega) = \begin{bmatrix} 0 & 2 \\ 0 & 2 \end{bmatrix} z_{L_{m+1}}. \] (55)

In filter design and control systems theory [3], the frequency response of a linear continuous-time invariant system is often obtained by substituting \( s = j\omega \) in the system transfer function \( H(s) = \frac{g(s)}{p(s)} \), as the frequency is varied between two fixed values, then \( g(j\omega) = \text{Re}\{g(j\omega)\} + j\text{Im}\{g(j\omega)\} \), \( p(j\omega) = \text{Re}\{p(j\omega)\} + j\text{Im}\{p(j\omega)\} \) and \( H(j\omega) = |H(j\omega)|e^{j\phi(\omega)} \) where the magnitude and phase of the system are given as
\[ |H(j\omega)|^2 = H(j\omega)H^*(-j\omega), \]
\[ = (\text{Re}\{g(j\omega)\})^2 + (\text{Im}\{g(j\omega)\})^2 \]
\[ (\text{Re}\{p(j\omega)\})^2 + (\text{Im}\{p(j\omega)\})^2, \] (56)

and
\[ \phi(\omega) = \tan^{-1} \left( \frac{\text{Re}\{p(j\omega)\}\text{Im}\{q(j\omega)\} - \text{Re}\{q(j\omega)\}\text{Im}\{p(j\omega)\}} {\text{Re}\{p(j\omega)\}\text{Re}\{q(j\omega)\} + \text{Im}\{q(j\omega)\}\text{Im}\{p(j\omega)\}} \right). \] (57)

Therefore, the real and the imaginary parts of the system frequency response are
\[ \text{Re}\{H(j\omega)\} = |H(j\omega)|\cos(\phi(\omega)) \] (58)

and
\[ \text{Im}\{H(j\omega)\} = |H(j\omega)|\sin(\phi(\omega)). \] (59)
The calculation of the real and imaginary parts of the system frequency response using the expressions (58) and (59) takes $2(n + 4)$ real multiplications and $2(n + 4)$ real additions/subtractions for each value of the frequency. However, the conventional method requires about $6(n + 1) + 2$ real multiplications and $2(n + 1)$ real additions/subtractions. For stability analysis, it is usually enough to have a rough sketch of the frequency response, since only the intersects on the real axis of the complex plane are to be calculated. However, for complex systems and system designs, the computation may be too tedious to handle and time consuming and sometimes it may be necessary to know more about the plot itself before the final sketch is made correctly. The equations (56) and (57) provide the frequency response magnitude and phase to be used for the Bode plots drawing. Similarly, the equations (58) and (59) furnish the frequency response real and imaginary parts to be used in Nyquist plot drawing. Furthermore, the frequency response real and imaginary parts can also be used in the product computation of a rational function with its complex conjugate which is often used in optimal control to determine the symmetric or square root locus and to evaluate the cost function of a time invariant continuous system.

Note that if a numerator and/or denominator of a transfer function are given as products of first order factors, as

$$G(s) = \prod_{l=1}^{m}(1 + T_{nl}s) \prod_{k=1}^{n}(1 + T_{nk}s), \quad \text{with} \quad n \geq m. \quad (60)$$

Where $T_{n1}, \ldots, T_{nm},$ and $T_{d1}, \ldots, T_{dn}$ are the system numerator and denominator positive constants respectively. The previous expression of the system transfer function can also be rewritten as

$$G(s) = K \sum_{\nu=0}^{m} q_\nu s^{m-\nu} \sum_{k=0}^{n} p_\nu s^{n-\nu}, \quad (61)$$

where the gain $K$ is found to be given as

$$K = \prod_{l=1}^{m} T_{nl} \prod_{k=1}^{n} T_{nk}, \quad (62)$$

and the numerator and denominator coefficients $q_0, \ldots, q_m$ and $p_0, \ldots, p_n$ respectively, are computed as follows

$$e_f = (-1)^f \sum_{k=1}^{g} \prod_{l=1}^{f} r_k. \quad (63)$$
Where
\[ g = \frac{n!}{f!(n-f)!} \]  \hspace{1cm} (64)
and
\[ t = \begin{cases} (k+l-1) \mod n & k+l \neq en + 1, \\ n & k+l = en + 1, \end{cases} \]  \hspace{1cm} (65)
for some integer e. The roots and the coefficients of the numerator and denominator of \( G(s) \) are then given respectively as
- Numerator
\[ r_f = -\frac{1}{T_{nf}}, \quad q_f = c_f, \quad \text{for } f = 1, \ldots, m. \]  \hspace{1cm} (66)
- Denominator
\[ r_f = -\frac{1}{T_{nf}}, \quad p_f = c_f, \quad \text{for } f = 1, \ldots, n. \]  \hspace{1cm} (67)

Usually, the system dynamics are given by the following state space representation [3]
\[ \dot{x}(t) = Ax(t) + Bu(t) \]  \hspace{1cm} (68)
\[ y(t) = Cx(t) \]  \hspace{1cm} (69)
where \( A \) is the \( n \times n \) state matrix, \( B \) is the \( n \times 1 \) input vector and \( C \) is the \( 1 \times n \) output vector. The system transfer function [3] is
\[ G(s) = C(sI - A)^{-1}B, \]  \hspace{1cm} (70)
\[ = \frac{1}{p(s)}C\text{adj}(sI - A)B, \]  \hspace{1cm} (71)
\[ = \frac{q(s)}{p(s)}, \]  \hspace{1cm} (72)
where \( p(s) = \sum_{l=0}^{n} p_l s^{n-l} \) is the system characteristic equation and \( q(s) = C\text{adj}(sI - A)B = \sum_{l=1}^{n} q_l s^{n-l} \). The matrix adjoint is given as
\[ \text{adj}(sI - A) = \sum_{l=1}^{n} F_l s^{n-l} \]  \hspace{1cm} (73)
where the matrices \( F_l^s \) and the coefficients \( p_l \) are computed using the Leverrier - Faddeva algorithm
\[ F_{l+1} = AF_l + p_l I, \]  \hspace{1cm} (74)
\[ p_l = -\frac{1}{l} \text{tr}(AF_l), \]  \hspace{1cm} (75)
with \( F_1 = I_n, \ F_{n+1} = 0, \ p_0 = 1 \) and \( \text{tr}(\cdot) \) is the trace of a matrix. The matrices \( F_l^s \) of equation (74) and the coefficients \( p_l \) of equation (75) can also be given by
\[ F_k = \sum_{l=0}^{k-1} p_l A^{k-l-1}, \]  \hspace{1cm} (76)
\[ p_k = -\frac{1}{k} \sum_{l=0}^{k-1} p_l \text{tr}(A^{k-l}), \]  \hspace{1cm} (77)
for \( 0 \leq k \leq n \). Thus, the transfer function numerator coefficients are found to be
\[ q_k = CF_k B = \sum_{l=0}^{k-1} p_l C A^{k-l-1} B. \]  \hspace{1cm} (78)
Notice that the use of equations (76), (77) and (78) requires the computation of a matrix to a certain power. The matrix exponentiation is very important subject in control, image and signal processing. Thus if the exponent of the matrix exponentiation is large enough, the calculation becomes tedious. Therefore, a technique of conversion from a binary number to an integer is used to develop a fast algorithm for the computation of a matrix exponentiation. The method of conversion is given as follows:

Let \( \rho_0, \rho_1, \ldots, \rho_{k-1} \) be the binary digits of the integer \( \rho \), such that

\[
(\rho_{k-1}\rho_{k-2} \cdots \rho_1\rho_0)_2 = (\rho)_{10}
\]

(79)

To begin with, one must discard the most significant digit \( \rho_{k-1} \), and replace every other digit remaining that follows starting from the left, by a multiplication of 2, if \( \rho_l = 0 \) and by a multiplication of 2 plus 1, if \( \rho_l = 1 \). The most significant digit remaining is associated with the most inner parenthesis of the expansion. For example \( 117_{10} = 1110101_2 \) gives

\[
117 = 1 + 2(1 + 2(1 + 2(1 + 2)))
\]

Based on the binary conversion given above, a fast algorithm for the computation of a matrix to a certain power is presented and it is given as follows:

To compute \( A^\rho \) where \( (\rho)_{10} = (\rho_{k-1}\rho_{k-2} \cdots \rho_1\rho_0)_2 \).

Let

\[
m_{k-i-1} = \begin{cases} 
m_{k-i}^2 & \text{if } \rho_{k-i-1} = 0 \\
A m_{k-i} & \text{if } \rho_{k-i-1} = 1 
\end{cases}
\]

(80)

for \( 1 \leq i \leq k - 1 \), where \( m_{k-1} = A \) and \( m_0 = A^\rho \).

In the coming section, the developed polynomial evaluation algorithms will be extended to the 2D polynomials and applied to the 2D continuous time systems.

### 8. TWO DIMENSIONAL POLYNOMIALS

The algorithms given in [4] and repeated in the previous sections will be extended to the two dimensional case. Although, a vast quantity of results has been found for the one variable polynomial, relatively fewer of these results can be extended to the two variables case. The most serious problem in the generalization of the one variable techniques to the two variable counterpart, is the fact that there is no fundamental theorem of algebra for polynomials in two independent variables.

Consider the linear continuous time invariant two dimensional (2D) system described by the transfer function
\[
G(s_1, s_2) = \frac{q(s_1, s_2)}{p(s_1, s_2)}
\]  
(81)

where

\[
q(s_1, s_2) = \sum_{l=0}^{n_1} \sum_{k=0}^{n_2} q_{lk} s_1^{n_1-l} s_2^{n_2-k} = S_1 QS_2^T,
\]  
(82)

the system characteristic polynomial is given as

\[
p(s_1, s_2) = \sum_{l=0}^{m_1} \sum_{k=0}^{m_2} p_{lk} s_1^{m_1-l} s_2^{m_2-k} = S_1 PS_2^T,
\]  
(83)

with

\[
S_1 = \begin{bmatrix}
s^{n_2}_1 & s^{n_2-1}_1 & \ldots & 1
\end{bmatrix}, \quad S_2 = \begin{bmatrix}
s^{m_2}_2 & s^{m_2-1}_2 & \ldots & 1
\end{bmatrix},
\]

\[
Q = \begin{bmatrix}
q_{00} & q_{01} & \cdots & q_{0,m_1} \\
q_{10} & q_{11} & \cdots & q_{1,m_1} \\
\vdots & \vdots & & \vdots \\
q_{n_1,0} & q_{n_1,1} & \cdots & q_{n_1,m_1}
\end{bmatrix}
\]

and

\[
P = \begin{bmatrix}
p_{00} & p_{01} & \cdots & p_{0,m_2} \\
p_{10} & p_{11} & \cdots & p_{1,m_2} \\
\vdots & \vdots & & \vdots \\
p_{m_1,0} & p_{m_1,1} & \cdots & p_{m_1,m_2}
\end{bmatrix}
\]

The variables \( s_1 \) and \( s_2 \) are the 2D polynomial variables along the horizontal and vertical directions, respectively.

If \( p(s_1, s_2) \) is separable (factorable), that is \( p(s_1, s_2) = p_1(s_1) p_2(s_2) \) and the matrix \( P \) has unit rank, then

\[
P = \begin{bmatrix}
\eta_0 \\
\eta_1 \\
\vdots \\
\eta_{m_2}
\end{bmatrix} \begin{bmatrix}
q_{0} & q_{1} & \cdots & q_{m_2}
\end{bmatrix},
\]

where \( \eta = \gcd(p_{n_1}), \eta_2 = \gcd(p_{n_2}) \), for \((0,0) \leq (l,k) \leq (n_2, m_2)\), then \( p_1(s_1) = \sum_{l=0}^{n_2} \eta_1 s_1^{n_1-l} \) and \( p_2(s_2) = \sum_{k=0}^{m_2} \eta_2 s_2^{m_2-k} \).

The algorithms developed in [4] and repeated in section 4, can be used twice in two different directions horizontally and vertically to evaluate the 2D polynomial \( p(s_1, s_2) \), whose magnitude would be the product of the individual magnitudes (horizontal and vertical), and its phase would be the sum of the individual phases.

If \( p(s_1, s_2) \) cannot be factored, then let

\[
p(s_1, s_2) = \sum_{k=0}^{m_2} f_k(s_1) s_2^{m_2-k},
\]  
(84)

where

\[
f_k(s_1) = \sum_{l=0}^{n_2} p_{lk} s_1^{n_2-1}.
\]  
(85)
Now, let $s_1 = a_1 + jb_1$ and $s_2 = a_2 + jb_2$ such that
\[ f_k(a_1 + jb_1) = \alpha_k + j\beta_k \]  \hspace{1cm} (86)
for $0 \leq k \leq m_2$, and $e_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.
\[ R_{e1} = \begin{bmatrix} 0 & 1 \\ -(a_1^2 + b_1^2) & 2a_1 \end{bmatrix}, \quad T_1 = \begin{bmatrix} -a_1 & 1 \\ b_1 & 0 \end{bmatrix}, \]
\[ R_{e2} = \begin{bmatrix} 0 & 1 \\ -(a_2^2 + b_2^2) & 2a_2 \end{bmatrix}, \quad T_2 = \begin{bmatrix} -a_2 & 1 \\ b_2 & 0 \end{bmatrix}, \]
and let $w_{0k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, then
\[ w_{l+1,k} = R_{e1}w_{lk} + pt_k e, \]  \hspace{1cm} (87)
for $0 \leq l \leq n_2$, and
\[ \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = T_l w_{n_2+1,k}. \]  \hspace{1cm} (88)

Hence,
\[ p(a_1 + jb_1, s_2) = \sum_{k=0}^{m_2} \alpha_k e_2^{m_2-k} + j \sum_{k=0}^{m_2} \beta_k e_2^{m_2-k} \]  \hspace{1cm} (89)
and
\[ p(a_1 + jb_1, a_2 + jb_2) = \eta_p + \vartheta_p. \]  \hspace{1cm} (90)

To compute $\eta_p$ and $\vartheta_p$, let $u_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and $v_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$,
\[ u_{k+1} = R_{e2}u_k + \alpha_k e, \]  \hspace{1cm} (91)
and
\[ v_{k+1} = R_{e2}v_k + \beta_k e. \]  \hspace{1cm} (92)

The substitution of the expressions for $\alpha_k$ and $\beta_k$, obtained from equation (88), into the equations (91) and (92) yields
\[ u_{k+1} = R_{e2}u_k + \begin{bmatrix} -a_1 & 1 \end{bmatrix} w_{m_2+1,k} e, \]  \hspace{1cm} (93)
and
\[ v_{k+1} = R_{e2}v_k + \begin{bmatrix} b_1 & 0 \end{bmatrix} w_{m_2+1,k} e. \]  \hspace{1cm} (94)
Thus,
\[
\begin{bmatrix}
\eta_P \\
\eta_F
\end{bmatrix} = T_2 v_{m_2+1} + J^T T_2 v_{m_2+1}.
\tag{95}
\]

The same procedure can be applied to the numerator of \(G(s_1, s_2)\) to obtain
\[
q(a_1 + jb_1, o_2 + jb_2) = \eta_q + j\sqrt{\eta_q}.
\tag{96}
\]

The algorithm will be established using equations (84) - (95).

Since
\[
G(s_1, s_2) = \frac{q(s_1, s_2)}{p(s_1, s_2)},
\]

and
\[
G(a_1 + jb_1, o_2 + jb_2) = |G(a_1 + jb_1, o_2 + jb_2)|e^{j\phi_0(a_1 + jb_1, o_2 + jb_2)},
\]

Let
\[
\phi_p(a_1 + jb_1, o_2 + jb_2) = \tan^{-1} \left( \frac{\eta_p}{\eta_p} \right),
\tag{97}
\]

and
\[
\phi_q(a_1 + jb_1, o_2 + jb_2) = \tan^{-1} \left( \frac{\eta_q}{\eta_q} \right),
\tag{98}
\]

then
\[
\phi_h(a_1 + jb_1, o_2 + jb_2) = \phi_q(a_1 + jb_1, o_2 + jb_2) - \phi_p(a_1 + jb_1, o_2 + jb_2)
= \tan^{-1} \left( \frac{\eta_p\eta_q - \eta_q\eta_p}{\eta_p\eta_q + \eta_q\eta_p} \right),
\tag{99}
\]

and
\[
|G(a_1 + jb_1, o_2 + jb_2)| = \sqrt{\eta_p^2 + \eta_q^2}.
\]

Therefore,
\[
\text{Re}\{G(a_1 + jb_1, o_2 + jb_2)\} = |G(a_1 + jb_1, o_2 + jb_2)| \cos (\phi_h(a_1 + jb_1, o_2 + jb_2)),
\]

and
\[
\text{Im}\{G(a_1 + jb_1, o_2 + jb_2)\} = |G(a_1 + jb_1, o_2 + jb_2)| \sin (\phi_h(a_1 + jb_1, o_2 + jb_2)).
\]

If the 2D system is described by the following state variable representation
\[
\begin{bmatrix}
\frac{\partial x^p(t_1, t_2)}{\partial t_1} \\
\frac{\partial x^p(t_1, t_2)}{\partial t_2}
\end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x^p(t_1, t_2) \\ x^h(t_1, t_2) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t_1, t_2)
\tag{100}
\]

and
\[
y(t_1, t_2) = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x^p(t_1, t_2) \\ x^h(t_1, t_2) \end{bmatrix},
\tag{101}
\]
where $A_{11}$ and $A_{22}$ are $n \times n$ and $m \times m$ state submatrices respectively, $B_1$ and $B_2$ are $n \times 1$ and $m \times 1$ input concatenated column vectors, and $C_1$ and $C_2$ are $1 \times n$ and $1 \times m$ output concatenated row vectors.

The system transfer function is then given as

$$
G(s_1, s_2) = \left[ \begin{array}{c} C_1 \\ C_2 \end{array} \right] \left[ \begin{array}{c} s_1 I_n - A_{11} \\ s_2 I_m - A_{22} \end{array} \right]^{-1} \left[ \begin{array}{c} B_1 \\ B_2 \end{array} \right],
$$

(102)

$$
= \frac{1}{p(s_1, s_2)} \sum_{l=0}^{n} \sum_{k=0}^{m} C F_{l,k} B s_1^{n-l} s_2^{m-k},
$$

(103)

$$
= \frac{q(s_1, s_2)}{p(s_1, s_2)}
$$

(104)

where the system characteristic polynomial follows as

$$
p(s_1, s_2) = \sum_{l=0}^{n} \sum_{k=0}^{m} p_{lk} s_1^{n-l} s_2^{m-k},
$$

(105)

with $p_{00} = 1$ and

$$
q(s_1, s_2) = \sum_{l=0}^{n} \sum_{k=0}^{m} q_{lk} s_1^{n-l} s_2^{m-k} = \sum_{l=0}^{n} \sum_{k=0}^{m} C F_{l,k} B s_1^{n-l} s_2^{m-k},
$$

(106)

that is $q_{lk} = C F_{l,k} B$. The matrices $F_{lk}$ and the coefficients $p_{lk}$ are determined using the following 2D Leuerrier - Faddeeva algorithm

$$
F_{l,k} = A^{10} F_{l-1,k} + A^{01} F_{l,k-1} + p_{l-1,k} I^{10} + p_{l,k-1} I^{01},
$$

(107)

$$
p_{lk} = -\frac{1}{l+k} t_{r}(A F_{l,k}).
$$

(108)

with $l + k \neq 0$, $0 \leq l \leq n$, $0 \leq k \leq m$, the initial conditions

$$
p_{00} = 1, \quad F_{00} = 0, \quad F_{01} = I^{01}, \quad F_{10} = I^{10},
$$

and the checking condition $A F_{nm} + p_{nn} I = 0$.

Where

$$
I^{10} = \left[ \begin{array}{c} I_n \\ 0 \\ 0 \end{array} \right], \quad I^{01} = \left[ \begin{array}{c} 0 \\ 0 \\ I_m \end{array} \right],
$$

(109)

and

$$
A^{10} = I^{10} A, \quad A^{01} = I^{01} A, \quad I^{10} = I^{10} B, \quad I^{01} = I^{01} B.
$$

(110)

Thus,

$$
A^{lk} = A^{10} A^{l-1,k} + A^{01} A^{lk-1}.
$$

(111)
The matrices $F_{ik}$ and the coefficients $q_{ik}$ and $p_{ik}$ can now be determined as

\[ F_{ik} = \sum_{i=0}^{l} \sum_{j=0}^{k} p_{ij} \left( A^{l-i-1,k-j-1} f^{10} + A^{l-i,k-j-1} f^{01} \right), \tag{112} \]

\[ q_{ik} = \sum_{i=0}^{l} \sum_{j=0}^{k} p_{ij} C \left( A^{l-i-1,k-j} B^{10} + A^{l-i-1,k-j-1} B^{01} \right), \tag{113} \]

and

\[ p_{ik} = -\frac{1}{l+k} tr \left( A F_{ik} \right) = -\frac{1}{l+k} tr \left( F_{ik} A \right) \]

\[ = -\frac{1}{l+k} tr \left( \sum_{i=0}^{l} \sum_{j=0}^{k} p_{ij} \left( A^{l-i-1,k-j-1} f^{10} + A^{l-i,k-j-1} f^{01} \right) A \right) \]

\[ = -\frac{1}{l+k} tr \left( \sum_{i=0}^{l} \sum_{j=0}^{k} p_{ij} \left( A^{l-i-1,k-j} A^{10} + A^{l-i,k-j-1} A^{01} \right) \right) \]

\[ = -\frac{1}{l+k} \left( \sum_{i=0}^{l} \sum_{j=0}^{k} p_{ij} tr \left( A^{l-i-1,k-j} \right) \right). \tag{114} \]

Similar algorithms can be developed for the discrete time systems by replacing the parameter $a$ of the previous algorithms with $\cos(\omega T)$ and the parameter $b$ with $\sin(\omega T)$, where $\omega$ is varied from $-\pi$ to $\pi$.

9. CONCLUSION

In this paper, we have introduced algorithms for the evaluation of complex polynomials with one and two variables which can used for the stability analysis of linear continuous time invariant systems. We have also presented a fast matrix exponentiation technique. These algorithms require fewer arithmetic operations than the existing techniques.

REFERENCES