DFT calculation of the generalized and drazin inverse of a polynomial matrix

N. P. Karampetakis[1] and S. Vologiannidis[2]
Department of Mathematics
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
email : karampet@auth.gr

Abstract

A new algorithm is presented for the determination of the generalized inverse and the drazin inverse of a polynomial matrix. The proposed algorithms are based on the discrete Fourier transform and thus are computationally fast in contrast to other known algorithms. The above algorithms are implemented in the Mathematica programming language and illustrated via examples.

1 Introduction

The first definition in the literature for a generalized inverse for nonsquare constant matrices is due to Penrose [11], while later Decell [2] proposed a Leverrier-Faddeev algorithm for its computation. A Leverrier-Faddeev algorithm has also been proposed by Grevile [4] for the computation of the Drazin inverse of square constant matrices. Karampetakis in [8] and later [7], [9] and [12] have proposed new Leverrier algorithms for the determination of the generalized inverse and Drazin inverse of polynomial matrices. These algorithms are good enough if we implement them in symbolic programming languages like Mathematica, Maple etc.. However their main disadvantage is the same with all the known Leverrier algorithms : are not stable if they are implemented in other high level programming languages such as C++, Fortran etc.

During the past two decades there has been extensive use of Discrete Fourier Transform (DFT) - based algorithms, due to their computational speed and accuracy. Some remarkable examples, but not the only, of the use of DFT in linear algebra problems are the calculation of the determinantal polynomial by [10], the computation of the transfer function of generalized n-dimensional systems by [1] and the solutions of polynomial matrix Diophantine equations by [6].

The reason for the interest in these two specific inverses are due to their applications in inverse systems, solution of AutoRegressive Moving Average representations [5] and solution of Diophantine equations which gives rise to numerous applications (see for example [8] and its references).

The main purpose of this work is to present a DFT-algorithm for the evaluation of the generalized inverse and the Drazin inverse of a polynomial matrix. More specifically in section 2 we introduce the 2-dimensional discrete Fourier transform, while later in section 3 and 4 we propose two new DFT algorithms for the evaluation of the generalized and Drazin inverse respectively of a polynomial matrix. Finally in section 5 we present a benchmark of the effectiveness of the algorithms implemented in Mathematica in comparison to the methods presented in section 6 of [9].

2 The discrete Fourier transform

Consider the finite sequence $X(k)$ and $\tilde{X}(r)$ $k,r = 0,1,...,M$. In order for the sequence $X(k)$ and $\tilde{X}(r)$ to constitute an DFT pair the following relations should hold [3] :

$$\tilde{X}(r) = \sum_{k=0}^{M} X(k)W^{-kr} \tag{1}$$

$$X(k) = \frac{1}{M+1}\sum_{r=0}^{M} \tilde{X}(r)W^{kr} \tag{2}$$

where

$$W = e^{\frac{2\pi j}{M+1}} \tag{3}$$

$X, \tilde{X}$ are discrete argument matrix-valued functions, with dimensions $p \times m$.

Consider now the finite sequence $X(k_1,k_2)$ and $\tilde{X}(r_1,r_2)$, $k_i, r_i = 0,1,...,M_i, i = 1,2$. In order for

[1]e-mail: karampet@auth.gr
[2]e-mail: svol@math.auth.gr

the sequence $X(k_1,k_2)$ and $\tilde{X}(r_1,r_2)$ to constitute an DFT pair the following relations should hold [3] :

$$\tilde{X}(r_1,r_2)=\sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2}X(k_1,k_2)W_1^{-k_1r_1}W_2^{-k_2r_2} \quad (4)$$

$$X(k_1,k_2)=\frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2}\tilde{X}(r_1,r_2)W_1^{k_1r_1}W_2^{k_2r_2} \quad (5)$$

where

$$R=(M_1+1)\times(M_2+1) \quad (6)$$
$$W_i=e^{\frac{2\pi j}{M_i+1}}, i=1,2$$

$X,\tilde{X}$ are discrete argument matrix-valued functions, with dimensions $p\times m$.

3 Generalized inverse of a polynomial matrix

Consider the polynomial matrix

$$A(s)=A_qs^q+\cdots=A_1s+A_0\in R[s]^{p\times m} \quad (7)$$

with $A_i\in R^{p\times m}, i\in q$, and p not necessarily equal to m.

Definition 1 *[11] For every matrix $A\in R^{p\times m}$, a unique matrix $A^+\in R^{m\times p}$, which is called generalized inverse, exists satisfying*

(i) $AA^+A=A$

(ii) $A^+AA^+=A^+$

(iii) $(AA^+)^T=AA^+$

(iv) $(A^+A)^T=A^+A$

where A^T denotes the transpose of A. In the special case that the matrix A is square nonsingular matrix, the generalized inverse of A is simply its inverse i.e. $A^+=A^{-1}$.

In an analogous way we define the generalized inverse $A(s)^+\in R(s)^{m\times p}$ of the polynomial matrix $A(s)\in R[s]^{p\times m}$ defined in (7) as the matrix which satisfies the properties (i)-(iv) of Definition 1. [8] proposed the following Theorem for the computation of the generalized inverse of a polynomial matrix (7).

Theorem 2 *[8] Let $A(s)\in R[s]^{p\times m}$ as in (7) and*

$$\begin{aligned}a(s,z)&=\det\left[zI_p-A(s)A(s)^T\right]\\&=a_0(s)z^p+a_1(s)z^{p-1}+\cdots+a_{p-1}(s)z+a_p(s)\end{aligned} \quad (8)$$

where

$$a_0(s)=1$$

be the characteristic polynomial of $A(s)A(s)^T$. Let $a_p(s)\equiv 0,...,a_{k+1}(s)\equiv 0$ while $a_k(s)\neq 0$ and $\Lambda:=\{s_i\in R: a_k(s_i)=0\}$. Then the generalized inverse $A(s)^+$ of $A(s)$ for $s\in R-\Lambda$ is given by

$$A(s)^+=-\frac{1}{a_k(s)}A(s)^TB_{k-1}(s) \quad (9)$$
$$B_{k-1}(s)=a_0(s)\left(A(s)A(s)^T\right)^{k-1}+\cdots+a_{k-1}(s)I_p$$

If $k=0$ is the largest integer such that $a_k(s)\neq 0$, then $A(s)^+=0$. For those $s_i\in\Lambda$ find the largest integer $k_i<k$ such that $a_{k_i}(s_i)\neq 0$ and then the generalized inverse $A(s_i)^+$ of $A(s_i)$ is given by

$$A(s_i)^+=-\frac{1}{a_{k_i}(s_i)}A(s_i)^TB_{k_i-1}(s_i)$$
$$B_{k_i-1}(s_i)=a_0(s)\left(A(s_i)A(s_i)^T\right)^{k-1}+\cdots+a_{k_i-1}(s_i)I_p$$

Although the above algorithm is good enough for a symbolic programming language, it is not the appropriate for a high level programming language. Therefore in the sequel we propose a 4 step algorithm for the computation of the generalized inverse of $A(s)$ through DFT transforms.

Algorithm 3 *(Evaluation of the generalized inverse of $A(s)$)*

Step 1. (Evaluation of the polynomial $a(s,z)$)

It is easily seen from (8), that the greatest power n_1 of s in $a(s,z)$ is equal to the greatest power among the powers of $a_i(s), i=1,2,..,p$. Note however, [8] that the greatest power of $a_k(s)$ is $2kq$ i.e. $n_1=\max\{2kq,k=1,2,...,p\}=2pq$. The greatest power n_2 of z in $a(s,z)$ is p i.e. $n_2=p$. Thus the polynomial $a(s,z)$ can be written as

$$a(s,z)=\sum_{l_1=0}^{n_1}\sum_{l_2=0}^{n_2}a_{l_1,l_2}s^{l_1}z^{l_2} \quad (10)$$

The polynomial $a(s,z)$ can be numerically computed using the following $R=(2pq+1)\times(p+1)$ points

$$u_i(r_j)=W_i^{-r_j}, i=1,2 \quad (11)$$
$$i=1,2 \text{ and } r_j=0,1,...,M_i$$

where

$$W_i = e^{\frac{2\pi j}{M_i+1}}$$
$$i = 1, 2 \ \ ; \ \ M_1 = 2pq \ \ ; \ \ M_2 = p$$

To evaluate the coefficients a_{r_1,r_2} define

$$\tilde{a}_{r_1,r_2} = \det[u_2(r_2)I_p - A(u_1(r_1))A(u_1(r_1))^T] \quad (12)$$

From equations (10), (11) and (12) it follows that

$$\tilde{a}_{r_1,r_2} = \sum_{l_1=0}^{n_1} \sum_{l_2=0}^{n_2} a_{l_1,l_2} W_1^{-r_1 l_1} W_2^{-r_2 l_2} \quad (13)$$

Using equations (13) and (5) it is obvious that $[\tilde{a}_{r_1,r_2}]$ and $[a_{l_1,l_2}]$ form a DFT pair. Therefore the coefficients $[a_{l_1,l_2}]$ can be computed using the inverse 2-D DFT as follows

$$a_{l_1,l_2} = \frac{1}{R} \sum_{r_1=0}^{n_1} \sum_{r_2=0}^{n_2} \tilde{a}_{r_1,r_2} W_1^{r_1 l_1} W_2^{r_2 l_2}$$

where $l_1 = 0, 1, ..., 2pq$ and $l_2 = 0, 1, .., p$.

Step 2. (Evaluate $a_k(s)$)

Find $k : a_{k+1}(s) = a_{k+2}(s) = \cdots = a_p(s) = 0$ and $a_k(s) \neq 0$ or $a_{l_1,0} = a_{l_1,1} = \cdots = a_{l_1,k+1} = 0 \ \forall l_1$ and $a_{l_1,k} \neq 0$ for some k.

Step 3. (Evaluate $A(s)^T B_{k-1}(s)$)

It is easily seen that the greatest power n of s in

$$B(s) = A(s)^T B_{k-1}(s) = A(s)^T \left[\left(A(s)A(s)^T\right)^{k-1} + \cdots + a_{k-1}(s)I_p \right]$$

is $n = \max\{2(k-1)q + q, k = 1, 2, ..., p\} = (2p-1)q$. Thus the polynomial matrix $B(s)$ can be written as

$$B(s) = \sum_{l=0}^{n} B_l s^l \quad (14)$$

The polynomial $B(s)$ can be numerically computed using the following $R = (2p-1)q + 1$ points

$$u(r) = W^{-r} \quad (15)$$

where

$$W = e^{\frac{2\pi j}{(2p-1)q+1}}$$

To evaluate the coefficients B_l define

$$\tilde{B}_r = B(u(r)) \quad (16)$$

From equations (14), (15) and (16) it follows that

$$\tilde{B}_r = \sum_{l=0}^{n} B_l W^{-lr} \quad (17)$$

Using equations (17) and (2) it is obvious that $[\tilde{B}_i]$ and $[B_l]$ form a DFT pair. Therefore the coefficients $[B_l]$ can be computed using the inverse DFT as follows

$$B_l = \frac{1}{R} \sum_{r=0}^{n} \tilde{B}_r W^{lr}$$

where $l = 0, 1, ..., (2p-1)q$.

Step 4. (Evaluation of the generalized inverse)

$$A(s)^+ = \frac{B(s)}{-a_k(s)}$$

4 Drazin inverse of a polynomial matrix

Using the same approach with the previous section, we define the Drazin inverse of a polynomial matrix and find a DFT algorithm for its computation.

Definition 4 *For every matrix $A \in R^{m \times m}$, a unique matrix $A^D \in R^{m \times m}$, which is called Drazin inverse, exists satisfying*

(i) $A^{k+1}A^D = A^k$ for $k = ind(A) = \min(k \in N : rank\left(A^k\right) = rank\left(A^{k+1}\right))$

(ii) $A^D A A^D = A^D$

(iii) $AA^D = A^D A$

In the special case that the matrix A is square nonsingular matrix, the Drazin inverse of A is simply its inverse i.e. $A^D = A^{-1}$.

In an analogous way we define the Drazin inverse $A(s)^D \in R(s)^{m \times m}$ of the polynomial matrix $A(s) \in R[s]^{m \times m}$ defined in (7) (with $p = m$) as the matrix which satisfies the properties (i)-(iii) of Definition 4. [12] proposed the following algorithm for the computation of the Drazin inverse of a polynomial matrix (7).

Theorem 5 *[12] Consider a nonregular one-variable rational matrix $A(s)$. Assume that*

$$a(z,s) = \det\left[zI_m - A(s)\right] = = a_0(s)z^m + a_1(s)z^{m-1} + \cdots a_{m-1}(s)z + a_m(s),$$

where

$$a_0(s) \equiv 1, \quad z \in \mathbb{C}$$

is the characteristic polynomial of $A(s)$. Also, consider the following sequence of $m \times m$ polynomial matrices

$$B_j(s) = a_0(s)A(s)^j + \cdots a_{j-1}(s)A(s) + a_j(s)I_m,$$
$$a_0(s) = 1, \quad j = 0, \ldots, m$$

Let

$$a_m(s) \equiv 0, \ldots, a_{t+1}(s) \equiv 0, \quad a_t(s) \neq 0.$$

Define the following set:

$$\Lambda = \{s_i \in \mathbb{C}\colon a_t(s_i) = 0\}$$

Also, assume

$$B_m(s) = \cdots = B_r(s) = 0, B_{r-1}(s) \neq 0$$

and $k = r - t$. In the case $s \in C\backslash\Lambda$ and $k > 0$, the Drazin inverse of $A(s)$ is given by

$$A(s)^D = (-1)^{k+1}a_t(s)^{-k-1}A(s)^k B_{t-1}(s)^{k+1}$$
$$B_{t-1}(s) = a_0(s)A(s)^{t-1} + \cdots + a_{t-1}(s)A(s)^0$$

In the case $s \in C\backslash\Lambda$ and $k = 0$, we get $A(s)^D = O$.

For $s_i \in \Lambda$ denote by t_i the largest integer satisfying $a_{t_i}(s_i) \neq 0$, and by r_i the smallest integer satisfying $B_{r_i}(s_i) \equiv O$. Then the Drazin inverse of $A(s_i)$ is equal to

$$A(s_i)^D = (-1)^{k_i+1}a_{t_i}(s_i)^{-k_i-1}A(s_i)^{k_i}B_{t_i-1}(s_i)^{k_i+1}$$
$$B_{t_i-1}(s_i) = a_0(s_i)A(s_i)^{t_i-1} + \cdots + a_{t_i-1}(s_i)A(s_i)^0$$

where $k_i = r_i - t_i$.

It is known that a q-th order polynomial is zero iff its value at $q+1$ points is zero. The same holds for polynomial matrices as we can easily in the following Lemma.

Lemma 6 *A polynomial matrix*

$$B(s) = B_0 + B_1 s + \cdots + B_q s^q \in R[s]^{m\times m}$$

is the zero polynomial matrix iff its value at $q+1$ distinct points is the zero matrix.

Proof: Consider that the value of the matrix $B(s)$ at the $q+1$ distinct points $s_i, i = 0, 1, ..., q+1$ is zero. Then we shall have that

$$\left[\begin{array}{cccc} B(s_0) & B(s_1) & \cdots & B(s_q) \end{array}\right] =$$
$$= \left[\begin{array}{cccc} B_0 & B_1 & \cdots & B_q \end{array}\right] \underbrace{\left[\begin{array}{cccc} I_m & I_m & \cdots & I_m \\ s_0 I_m & s_1 I_m & \cdots & s_q I_m \\ \vdots & \vdots & \ddots & \vdots \\ s_0^q I_m & s_1^q I_m & \cdots & s_q^q I_m \end{array}\right]}_{R}$$
$$= 0_{[m]\times[(q+1)m]}$$

However since the $q+1$ points are distinct the Vandermode matrix R has nonzero determinant and therefore the above system of equations has the unique solution $B_i = 0, i = 0, 1, ..., q$. ■

In what follows, we propose a 6-step algorithm for the evaluation of the Drazin inverse of a polynomial matrix $A(s)$.

Algorithm 7 *(Evaluation of the Drazin inverse of a polynomial matrix $A(s)$)*

Step 1. (Evaluation of the polynomial $a(s,z)$)

It is easily seen that the greatest power n_1 of s in $a(s,z)$ is equal to the greatest power among the powers of $a_i(s), i = 1, 2, .., m$. Note [9] that the greatest power of $a_k(s)$ is $2kq$ i.e. $n_1 = \max\{2kq, k = 1, 2, ..., m\} = 2mq$. The greatest power n_2 of z in $a(s,z)$ is m i.e. $n_2 = m$. Thus the polynomial $a(s,z)$ can be written as

$$a(s,z) = \sum_{l_1=0}^{n_1}\sum_{l_2=0}^{n_2} a_{l_1,l_2} s^{l_1} z^{l_2} \tag{18}$$

The polynomial $a(s,z)$ can be numerically computed using the following $R = (2mq+1)\times(m+1)$ points

$$u_i(r_i) = W_i^{-r_i}, i = 1, 2 \tag{19}$$

where

$$W_i = e^{\frac{2\pi j}{M_i+1}}$$
$$i = 1, 2 \ ; \ M_1 = 2mq \ ; \ M_2 = m$$

To evaluate the coefficients a_{l_1,l_2} define

$$\tilde{a}_{r_1,r_2} = \det[u_2(r_2)I_m - A(u_1(r_1))] \tag{20}$$

From equations (18), (19) and (20) it follows that

$$\tilde{a}_{r_1,r_2} = \sum_{l_1=0}^{n_1}\sum_{l_2=0}^{n_2} a_{l_1,l_2} W_1^{-r_1 l_1} W_2^{-r_2 l_2} \tag{21}$$

Using equations (21) and (5) it is obvious that $[\tilde{a}_{r_1,r_2}]$ and $[a_{l_1,l_2}]$ form a DFT pair. Therefore the coefficients $[a_{l_1,l_2}]$ can be computed using the inverse 2-D DFT as follows

$$a_{l_1,l_2} = \frac{1}{R}\sum_{r_1=0}^{n_1}\sum_{r_2=0}^{n_2} \tilde{a}_{r_1,r_2} W_1^{r_1 l_1} W_2^{r_2 l_2}$$

where $l_1 = 0, 1, ..., 2mq$ and $l_2 = 0, 1, .., m$.

Step 2. (Evaluate $a_t(s)$)

Find t : $a_{t+1}(s) = a_{t+2}(s) = \cdots = a_m(s) = 0$ and $a_t(s) \neq 0$ or $a_{r_1,0} = a_{r_1,1} = \cdots = a_{r_1,t+1} = 0 \ \forall r_1$ and $a_{r_1,t} \neq 0$ for some t.

Step 3. (Evaluate $r \geq t$: $B_m(s) \equiv 0,...,B_r(s) \equiv 0, B_{r-1}(s) \neq 0$)

Now using lemma 6, we can easily find an algorithm for the determination of r. More specifically consider the polynomial matrix

$$B_j(s) = A(s)^j + a_1(s)A(s)^{j-1} + \cdots a_{j-1}(s)A(s) + a_j(s)I_m$$

The greatest power n of s in $B_j(s)$ is $n_j = 2jq$. In order now to determine the value of $r \geq t$ which satisfy the property : $B_m(s) \equiv \nvdash,...,B_r(s) \equiv 0, B_{r-1}(s) \neq 0$ we use the following short algorithm :

$j = m$

Determine the value of $B_j(s)$ at the following $n_j + 1$ points (or any other $n_j + 1$ distinct points)

$$u(r) = W^{-r}, W = e^{\frac{2\pi j}{n_j+1}}$$

Do WHILE ($B_j(s) = 0 \ \forall u(r)$)

$j = j - 1$

Determine the value of $B_j(s)$ at the following $n_j + 1$ points

$$u(r) = W^{-r}, W = e^{\frac{2\pi j}{n_j+1}}$$

END DO

$r = j$

The scepticism of the above short algorithm is that the polynomial matrix $B_j(s)$ of degree n_j coincides with the zero matrix if its value at $n_j + 1$ points is equal to zero (Lemma 6).

Step 4. (Evaluation of $A(s)^k B_{t-1}(s)^{k+1}$)

Let $k = r - t$. Then the greatest power n of s in

$$B(s) = A(s)^k B_{t-1}(s)^{k+1}$$
$$B_{t-1}(s) = a_0(s)A(s)^{t-1} + \cdots + a_{t-1}(s)A(s)^0$$

is $n = 2(t-1)q(k+1) + qk$. Thus the polynomial matrix $B(s)$ can be written as

$$B(s) = \sum_{l=0}^{n} B_l s^l \tag{22}$$

The polynomial matrix $B(s)$ can be numerically computed using the following $R = n + 1$ points

$$u(r) = W^{-r} \tag{23}$$

where

$$W = e^{\frac{2\pi j}{n+1}}$$

To evaluate the coefficients B_l define

$$\tilde{B}_r = B(u(r)) \tag{24}$$

From equations (22), (23) and (24) it follows that

$$\tilde{B}_r = \sum_{l=0}^{n} B_l W^{-lr} \tag{25}$$

Using equations (25) and (2) it is obvious that $[\tilde{B}_i]$ and $[B_l]$ form a DFT pair. Therefore the coefficients $[B_l]$ can be computed using the inverse DFT as follows

$$B_l = \frac{1}{R} \sum_{r=0}^{n} \tilde{B}_l W^{lr}$$

where $l = 0, 1, ..., n$.

Step 5. (Evaluation of $a_t(s)^{k+1}$)

The greatest power n of s in

$$a(s) = a_t(s)^{k+1}$$

is $n = 2tq(k+1)$. Thus the polynomial $a(s)$ can be written as

$$a(s) = \sum_{l=0}^{n} a_l s^l \tag{26}$$

The polynomial $a(s)$ can be numerically computed using the following $R = n + 1$ points

$$u(r) = W^{-r} \tag{27}$$

where

$$W = e^{\frac{2\pi j}{n+1}}$$

To evaluate the coefficients a_l define

$$\tilde{a}_r = a(u(r)) \tag{28}$$

From equations (26), (27) and (28) it follows that

$$\tilde{a}_r = \sum_{l=0}^{n} a_l W^{-lr} \tag{29}$$

Using equations (29) and (2) it is obvious that $[\tilde{a}_l]$ and $[a_l]$ form a DFT pair. Therefore the coefficients $[a_l]$ can be computed using the inverse DFT as follows

$$a_l = \frac{1}{R} \sum_{r=0}^{n} \tilde{a}_r W^{lr}$$

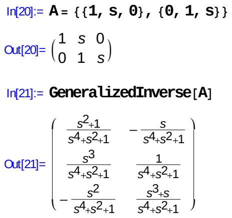
where $l = 0, 1, ..., n$.

Step 6. (Evaluation of the Drazin inverse)

$$A(s)^D = \frac{B(s)}{a(s)}$$

5 Implementation

In this section we will briefly describe the implementation of algorithm 3 and algorithm 7 in Mathematica. Note that the Mathematica code is available by contacting one of the authors. The Mathematica functions are *GeneralizedInverse[A]* and *DrazinInverse[A]* where A is a polynomial matrix with indeterminate s. A screenshot from a Mathematica notebook can be seen in the following figure.



The efficiency of the algorithms have been evaluated using the Mathematica function "Timing" which returns the CPU time consumed in seconds. Additionally, our implementation of Generalized (Drazin) inverse is compared with the implementation of [12] ([8]). For the test we used random matrices up to dimension 5 and degree 5. For both inverses the results were almost identical. Due to lack of space we present the comparison between the generalized inverse algorithms. The following graph shows the dependence of the CPU time consumed for the computation of the generalized inverse, versus the number of rows and the degree of the involved polynomial matrix. The red surface is the DFT based algorithm and the green one is the implementation of [8].

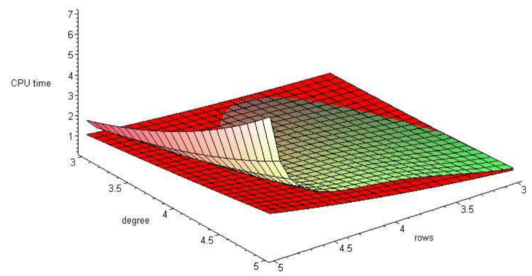


Diagram 1. Graph of the CPU time=F(rows,degree) in both algorithms DFTGI and [8].

It can be seen that for small values of the degree and the size of the polynomial matrix, the algorithm presented on [8] is better, while for bigger values of the degree and the size the DFT based algorithm gives better results. All the tests run on a COMPAQ Presario with CPU a Pentium III at 700MHz and 128Mb of memory, using Windows 2000 Professional and Mathematica 4.1.

6 Conclusions

In this paper we have presented two new algorithms for the computation of the generalized inverse and Drazin inverse of a polynomial matrix. The proposed algorithms have been tested and were shown to be more efficient from the known ones of [8] and [12] in the case where the degree and the size of the polynomial matrix gets bigger. The proposed algorithms can be extended to multivariable polynomial matrices. The clear benefit of computing such a generalized (Drazin) inverse is that it enables a wider set of such problems to be solved [7].

References

[1] G. E. Antoniou, 2001, Transfer function computation for generalized n-dimensional systems, Journal of the Franklin Institute, 338, 83-90.

[2] H.P. Decell, An application of the Cayley-Hamilton theorem to generalized matrix inversion, SIAM Review, 7, No 4, 1965, 526–528.

[3] D.E. Dudgeon, R.M. Mersereau, 1984, Multidimensional Digital Signal Processing, Prentice-Hall, Englewood Cliffs, N.J..

[4] T.N.E. Grevile, The Souriau-Frame algorithm and the Drazin pseudoinverse, Linear Algebra Appl., 6, 1973, 205–208.

[5] T. Guyer, O. Kiymaz, G. Bilgici and S. Mirasyedioglu, 2001, A new method for computing the solutions of differential equation systems using generalized inverse via Maple, Applied Mathematics and Computation, 121, 291-299.

[6] M. Hromcik and M. Sebek, 2001, Fast Fourier transform and linear polynomial matrix equations, Proceedings of the 1rst IFAC Workshop on Systems Structure and Control, Prague, Czech Republic.

[7] J. Jones, N.P. Karampetakis and A.C. Pugh, The computation and application of the generalized inverse via Maple, J. Symbolic Computation, 25, 1998, 99–124.

[8] N.P. Karampetakis, 1997, Computation of the generalized inverse of a polynomial matrix and applications., Linear Algebra and its Applications, 252, 35-60.

[9] N. Karampetakis and P. Stanimirovic, 2001, On the computation of the Drazin inverse of a polynomial matrix, 1rst IFAC Symposium on Systems Structure and Control, Prague, Czech Republic.

[10] L. E. Paccagnella and G. L. Pierobon, 1976, FFT calculation of a determinantal polynomial, IEEE Trans. on Automatic Control, pp.401-402.

[11] R. Penrose, 1955, A generalized inverse for matrices, Proc. Cambridge Philos. Soc., 51, 406-413.

[12] P. Stanimirovic and N.P. Karampetakis, 2000, Symbolic implementation of Leverrier-Faddeev algorithm and applications., 8th IEEE Medit. Conference on Control and Automation, Patra, Greece.