BRIEF PAPER

ALGORITHMS FOR FACTORIZATION OF THE IRREGULAR MATRIX POLYNOMIALS USING SYMBOLIC COMPUTATIONS

LATIFA F. AGAMALIEVA[1], FIKRET A. ALIEV[1], NAILE I. VELIEVA[1]

ABSTRACT. In this paper, the algorithm including symbolic computations of the package MATLAB Symbolic Toolbox is developed, for factorization of the irregular matrix polynomials. Software is given for the division of the polynomials, which admits symbolic computations. The proposed algorithm allows one to obtain high accuracy results.

Keywords: factorization, symbolic calculations, irregular polynomial.

AMS Subject Classification: 65Y04.

1. INTRODUCTION

Recently, with the use of new computing technologies and development of the modern software packages (including the Symbolic Toolbox MATLAB) the accuracy of solutions for some classes of problems significantly increased [3, 4]. For example, in [2, 6, 7] high accuracy algorithms are given for the solution of the matrix algebraic Riccati (MARE) and Lyapunov equations using Symbolic Toolbox of MATLAB. Using this approach in [8] high accuracy algorithm is developed for the factorization of the polynomial with scalar coefficients.

When implementing operations on numeric values a rounding error arises, since the accuracy of calculations is limited by the number of digits used in each operation. Therefore, under repeated operations errors are accumulated. Operations on symbolic variables can be implemented exactly as in these cases computations with numbers are not carried out, so the rounding error does not appear. MATLAB system carries out the calculations only in floating-point arithmetic. These operations are limited to the bit, which also in turn leads to errors. However, development and use of high technologies generates more classes of problems that require the computations without any errors. To achieve this, package of Symbolic Math Toolbox included in MATLAB is used.

It should be noted that there is a class of problems whose solution is reduced to the factorization of polynomials, and the separation of fractional-rational expressions. For example, solution of the Linear-Quadratic-Gauss (LQG) problems in the frequency domain is reduced to the solution of the Wiener-Hopf equation. And the last is closely related to the above procedures. It is known [1, 8] that the factorization of polynomials is reduced to solving MARE and for its solution high accuracy algorithms have been developed. In the paper, using these algorithms

[1]Institute of Applied Mathematics, Baku State University, Baku, Azerbaijan,
e-mail: latifa.agamalieva@gmail.com, f_aliev@yahoo.com, nailavi@rambler.ru
Manuscript received September, 2012.

high-accuracy computational procedures are proposed for factorization of the irregular matrix polynomials based on the symbolic computations.

2. Problem statement and calculation algorithm

Let the irregular matrix polynomial dimension $m \times m$ be given

$$B(s) = (-1)^n B_0 s^{2n} + B_1 s^{2n-1} + ... + B_{2n}, \tag{1}$$

where $B_i = B_i'$ for odd values of index and $B_i = -B_i'$ for even values.

It needs to factorize (1) i.e. to define a matrix $D(s)$ dimension $m \times m$ satisfying to the relation

$$B(s) = D_*(s)D(s), \tag{2}$$

where

$$D(s) = D_0 s^n + D_1 s^{n-1} + ... + D_n.$$

Here *- means the operation of transpose and replacing s by –s.

Note that the problem (1), (2) simplifies when B_0 has inverse.

We consider the case when B_0 has no inverse, i.e. the polynomial (1) is irregular.

For the factorization of the irregular matrix polynomial one has to do the following steps [1]:

1. Multiplying the matrix $B(s)$ from right and left by the constructed in special form polynomial matrices $T_*(s)$, $T(s)$, transform it into the regular one $A(s) = T_*(s)B(s)T(s)$.

2. Factorizing the polynomial $A(s)$ by using the algorithm given in [8] obtain

$$A(s) = H_*(s) \cdot H(s),$$

where

$$H(s) = H_0 s^n + H_1 s^{n-1} + ... + H_n.$$

3. The seeking matrix is defined by the formula $D(s) = H(s)T^{-1}(s)$.

Let us illustrate the scheme of realization of the first step.

For the realization of this step the standard procedure svd (singular vector decomposition) from the MATLAB Symbolic Toolbox is used

$$[U, D, V] = svd(A),$$

where $A = U D V'$.

Multiplying $B(s)$ by V, we get the following polynomial

$$V'B(s)V = [B_{ij}(s)], \quad ij = 1, ..., m.$$

Then the matrix $T_1(s)$ is defined in the form

$$T_1(s) = diag\{1, ..., 1, \ (s + \alpha_1)^{\delta_1}, 1, ..., 1\},$$

where δ_m is the maximal order of the out off diagonal polynomial. After finite time of cycles we come to the regular polynomial

$$\bar{B}(s) = T_{0*}(s)B(s)T_0(s) = (-1)^n \bar{B}_{01} s^{2n} + ... + \bar{B}_{02n}. \tag{3}$$

Here the matrix $\bar{B}_{01}$ is positively defined, i.e. it may be presented in the form $\bar{B}_{01} = Q'Q$. For this purpose the procedure chol (A) (Cholesky decomposition) from the Symbolic Toolbox of Matlab is used, where Q is upper triangle matrix. Multiplying $\bar{B}(s)$ from the left by $(Q')^{-1}$ and

from the right by Q^{-1} one obtain the matrix polynomial with the unit matrix as a coefficient at the high order

$$\bar{B}(s) = (-1)^n E s^{2n} + A_1 s^{2n-1} + ... + A_{2n}, \qquad (4)$$

where $A_j = (Q')^{-1} \bar{B}_{0j} Q^{-1}, \ \ j = 1, 2, ..., 2n$.

Then the second step of the algorithm is executed. The matrix $H(s)$ is defined factorizing the polynomial (4). In this step the matrix algebraic Riccati equation (MARE) is solved using the matrix signum function [1, 5] method, which is realized by symbolic computations with arbitrary accuracy. We briefly describe the procedure of calculation the matrix of the signum function, which according to [5] is defined as

$$sgn A = \lim_{i \to \infty} A_i,$$
$$A_0 = A,$$
$$A_{i+1} = \tfrac{1}{2} \left(A_i + A_i^{-1} \right), \ \ i = 0, 1, ...$$

In the third step the seeking matrix $D(s)$ is defined

$$D(s) = H(s) Q T_j^{-1}(s) V_\nu' \cdots T_1^{-1}(s) V_1'.$$

Here the operation of multiplication the matrix from the right by the $T_j^{-1}(s)$ in fact consists of the division of the polynomials of the corresponding columns of this matrix by the polynomial $(s+\alpha_j)^{\delta_j}$ standing for the diagonal of the matrix $T_j(s)$. For the sake of simplicity of the computing procedure let us take $\alpha_j = 1$.

The package MATLAB includes the standard procedures of division and multiplication of polynomials. The arguments of these operations are numerical for variables. These procedures do not provide the results with the necessary accuracy, and do not support the Symbolic Toolbox of MATLAB. Considering this we developed software for multiplication and division of polynomials that supports Symbolic Toolbox of MATLAB.

Division of polynomials is reduced to sequential repetition of the following procedures:

1) In the first step a senior term of the dividend, and later, a senior member of the next difference is divided by a senior term of the divisor;

2) The result of the division gives another term in the quotient, which the divisor is multiplied. The obtained product is written under the dividend or the next difference;

3) If the degree of the obtained polynomial is greater or equal to the degree of the divisor, then steps 1, 2 are applied to it. Otherwise the procedure of division is completed. The last difference is indeed a remainder.

The body of the software for the division of polynomials using Symbolic Toolbox of MATLAB is given in the end of the paper.

Let's illustrate the realization of the algorithm of factorization of irregular matrix polynomial on the example (example 4.2, [1])

Example. It needs to factorize the following matrix polynimial

$$B(s) = \left[\begin{array}{cc} 3 - s^2 & s^4 \\ s^4 & 2 + s^8 \end{array} \right].$$

For this case in (1) we have

$$B_1 = B_2 = B_3 = B_5 = B_7 = 0,$$

$$B_0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, B_4 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, B_6 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, B_8 = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}.$$

The matrix $U = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ transforms B_0 into the diagonal form. Multiplying the polynomial $B(s)$ by U from the left and right one can obtain

$$U'B(s)U = \begin{bmatrix} 2 + s^8 & -s^4 \\ -s^4 & 3 - s^2 \end{bmatrix}.$$

The matrix $T(s)$ is defined in the form $T(s) = diag\{1, (s+1)^3\}$.
Calculating the matrix polynomial

$$B_1(s) = T_*(s)U'B(s)UT(s) = \begin{bmatrix} 2 + s^8 & -s^4(1+s)^3 \\ -s^4(1-s)^3 & (3-s^2)(1-s^2)^3 \end{bmatrix},$$

we have the following regular matrix polynomial

$$A(s) = A_1 s^8 + A_2 s^7 + A_3 s^6 + A_4 s^5 + A_5 s^4 + A_6 s^3 + A_7 s^2 + A_8 s + A_9 ,$$

where

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 3 \\ 3 & -6 \end{bmatrix}, A_4 = \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix}, A_5 = \begin{bmatrix} 0 & 1 \\ 1 & 12 \end{bmatrix},$$

$$A_6 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, A_7 = \begin{bmatrix} 0 & 0 \\ 0 & -10 \end{bmatrix}, A_8 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, A_9 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}.$$

Factorization $A(s) = H_*(s)H(s)$ gives

$$H_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$H_1 = \begin{bmatrix} 2.67024675215584066665425481245717 & -0.21388401189504250760922278296807 \\ 0.786115988104957492390777217033342 & 4.71879423709054869378898861900026 \end{bmatrix},$$

$$H_2 = \begin{bmatrix} 3.87409803207652462051118370777708 & -0.64165203568512752282766834890364 \\ 0.780048541938744817118803110698332 & 8.15638271127164608136696585700098 \end{bmatrix},$$

$$H_3 = \begin{bmatrix} 3.29907620009420317977232360999326 & -0.64165203568512752282766834890411 \\ 0.512135308959983613113063543559074 & 6.15638271127164608136696585700090 \end{bmatrix},$$

$$H_4 = \begin{bmatrix} 1.40338961790271231746155148175531 & -0.21388401189504250760922278296781 \\ 0.174635564427407292237837609196922 & 1.71879423709054869378898861900027 \end{bmatrix}$$

The accuracy of these calculations is $3.224839186938058e-029$.
Thus the seeking matrix $D(s)$ from (2) indeed has a form

$$D_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, D_1 = \begin{bmatrix} 0 & 2.67024675215584066665425481245717 \\ 0 & 0.786115988104957492390777217033342 \end{bmatrix},$$

$$D_2 = \begin{bmatrix} 0 & 3.87409803207652462051118370777708 \\ 0 & 0.780048541938744817118803110698332 \end{bmatrix},$$

$$D_3 = \begin{bmatrix} 0 & 3.2990762000942031797723236099326 \\ 1 & 0.5121353089599836131130635435974 \end{bmatrix},$$

$$D_4 = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix},$$

where

$$d_{11} = 0.21388401189504250760922278296807$$
$$d_{12} = 1.4033896179027123174615514817531$$
$$d_{21} = -1.7187942370905486937889886190026$$
$$d_{22} = 0.17463556442740729223783760919692$$

The coefficients of the irregular matrix polynomial obtained in result of multiplication $\bar{B}(s) = D_*(s)D(s)$ differ from the initial ones in the 29th digit. Thus the considered problem is solved with accuracy 1.577822734559815e-029.

Body of the software for the division of polynomials using Symbolic Toolbox:

```
function [Y, Zf] = bolme1(B, A)
syms Y Zf fir fi firr firout
A1=vpa(A);B1=vpa(B);
if ((size(A,1)>1 & size(A,2)>1) |length(size(A)) == 3 )
error('A must be a vector.')
end
if ((size(B,1)>1 & size(B,2)>1) |length(size(B)) == 3 )
error('B must be a vector.')
end
if(A(1)==0.)
error('The first element of A can not be zero.');
end
A=vpa(A,64);B=vpa(B,64);bir=vpa(1,128);y=0;firr=vpa([0]);
while y==0
nb=length(B); na=length(A); X= vpa([bir zeros(1,nb-na)]);
if( double(A(1))~=1)
B = vpa(B/A(1)); A = vpa(A/A(1));
end
if length(B) < length(A)
y=3;
end
if(length(B)>=length(A))
if (size(A,1)>1)
Apad = vpa([A; zeros(length(B)-length(A), 1)]);
else
Apad =vpa ([A, zeros(1, length(B)-length(A)) ],64);
Bpad=vpa(B,64);
end
fir = vurma(vpa(Apad),vpa(B(1)));fi=vurma(vpa(X),vpa(B(1)));
firr=polysum(vpa(firr),vpa(fi));
```

```
firOut=vpa(polysum(vpa(Bpad),-vpa(fir)),64);
for i=1:length(firOut)
if double(firOut(i))~=0.
k=i;
break
end
end
kk=0;firOut1=[];
for i=k:length(firOut)
kk=kk+1; firOut1(kk)=vpa(firOut(i),64);
end
firOut=vpa(firOut1,64);B=vpa(firOut,64);
end
end
Y=vpa(firr,64);Zf=vurma(vpa(firOut),vpa(A1(1)));
Zf=vpa(Zf,64).
```

References

[1] Aliev, F.A., Bordyug, B.A., Larin, V.B., (1991), H_2 Optimization and State Space Method for the Synthesis of Optimal Regulators, Baku, Elm, 274p.

[2] Aliev, F.A., Gasimov, Y.S., Velieva, N.I., Safarova N.A., Agamalieva L.F., (2012), High accuracy algorithms to the solution of the optimal output feedback problem for the linear systems, Proceedings of the Romanian Academy, Series A, 13(3), pp.207-214.

[3] Danilov, A.I., (2002), Computer Practicum on the Control Theory, Simulink-simulation in MATLAB, .:SUEE, 128p.

[4] Gorbachenko, V.I., (2011), Calculation Linear Algebra with Examples in MATLAB, BHV-Petersburg, 320p. http://www.bhv.ru/books/full_contents.php?id=189162)

[5] Kenney, C.S., Laub, A.J., (1995), The matrix sign function, IEEE Trans. Autom.Contr., 2(8), pp.1330-1348.

[6] Larin, V.B., (2007), High-accuracy algorithms for solution of discrete periodic Riccati equations., Appl. Comput. Math., 6(1), pp.10-17.

[7] Varga, A., (1995), On computing high accuracy solutions of a class of Riccati equations, Control-Theory and Advanced Technology, 10(4), part 5, pp.2005-2016.

[8] Velieva, N.I., Agamalieva, L.F., (2011), High-accuracy algorithm to the factorization of the polynomials in MATLAB, Transaction of ANAS, XXX(6), pp.128-133.

Latifa Agamalieva was born in February 1971 in Baku. She graduated from Baku State University. Presently she is a junior researcher at the Institute of Applied Mathematics Baku State University and Ph.D student at the Institute of Cybernetics of ANAS. Her research interests are in the field of numerical methods of the control theory.

Fikret A. Aliev, for a photograph and biography, see TWMS J. Pure Appl. Math., V1, N.1, 2010.

Naila Velieva was born in 1953 in Zengilan district of Azerbaijan (now occupied by Armenia). She graduated from Baku State University in 1975 and got her Ph.D degree in 2000. Now she is with the Institute of Applied Mathematics Baku State University as a vice-director. She has published more than 20 papers. Her research interests are in the fields of numerical methods, control theory and computer sciences.