ПРОГРАММИРОВАНИЕ

MSC 93C05 **DOI: 10.14529/mmp140306**

SOLVING OF A MINIMAL REALIZATION PROBLEM IN MAPLE

V.M. Adukov, South Ural State University, Chelyabinsk, Russian Federation, adukovvm@susu.ac.ru,
A.S. Fadeeva South Ural State University, Chelyabinsk, Russian Federation, fadeevaas@susu.ac.ru

In the computer algebra system Maple, we have created a package MinimalRealization to solve the minimal realization problem for a discrete-time linear time-invariant system. The package enables to construct the minimal realization of a system starting with either a finite sequence of Markov parameters of a system, or a transfer function, or any non-minimal realization. It is designed as a user library and consists of 11 procedures: ApproxEssPoly, ApproxNullSpace, Approxrank, ExactEssPoly, FractionalFactorizationG, FractionalFactorizationMP, MarkovParameters, MinimalityTest, MinimalRealizationG, MinimalRealizationMP, Realization2MinimalRealization. The realization algorithm is based on solving of sequential problems: (1) determination of indices and essential polynimials (procedures ExactEssPoly, ApproxEssPoly), (2) construction of a right fractional factorization of the transfer function (FractionalFactorizationG, FractionalFactorizationMP), (3) construction of the minimal realization by the given fractional factorization (MinimalRealizationG, MinimalRealizationMP, Realization2MinimalRealization). We can solve the problem both in the case of exact calculations (in rational arithmetic) and in the presence of rounding errors, or for input data which are disturbed by noise. In the latter case the problem is ill-posed because it requires finding the rank and the null space of a matrix. We use the singular value decomposition as the most accurate method for calculation of the numerical rank (Approxrank) and the numerical null space (ApproxNullSpace). Numerical experiments with the package MinimalRealization demonstrate good agreement between the exact and approximate solutions of the problem.

Keywords: discrete-time linear time-invariant systems; fractional factorization; minimal realization; algorithms for solving of realization problem.

Introduction

In this paper we describe the package MinimalRealization that is destined for solving of a minimal realization problem. The problem is one of a fundamental problem in linear system theory. Our package is a Maple implementation of results of the work [1], where a new algorithm for solving of the minimal realization problem was suggested. The algorithm is based on notions of indices and essential polynomials for a matrix sequence [2].

The system of computer algebra Maple was chosen for the implementation of the algorithm because it allows to solve the problem exactly if input data are rational numbers. This enables us to construct model examples for testing of algorithms for approximate solving of the problem. All routines of the package are designed in such a way that the problem is solved exactly if input data are rational numbers, and algorithms of approximate solving are used if the data are floating-point numbers.

In the presence of rounding errors or for noisy input data the minimal realization problem is ill-posed because it uses ranks and null spaces of matrices. We apply the singular

value decomposition (SVD) as the most accurate method for calculation of the numerical rank and the numerical null spaces. An application of SVD to the problem was described in the review [3]. There are other approximate methods for solving of the realization problem. For example, a series of works of S.G. Pushkov and coauthors devoted to interval computations in the realization problem (see [4]).

Numerical experiments with the package MinimalRealization showed a good correspondence between the exact and approximate solutions of the problem.

1. Setting of the Problem

There are the various settings of the minimal realization problem (see, e.g., [3, 5, 6]). From a formal point of view, they can be reduced to the following three problems.

1.1. Construction of the Minimal Realization by the Impulse Response (by the Sequence of the Markov Parameters) of a System

Given a sequence of real or complex $p \times q$ matrices $\{G_k\}_{k=1}^{\infty}$ (*the impulse responce* or *the sequence of the Markov parameters of a system*). Find a triplet $\Sigma = (A, B, C)$ of matrices such that

$$CA^{k-1}B = G_k, \quad k = 1, 2, \ldots . \tag{1}$$

We call Σ the system. The sequence $\{G_k\}_{k=1}^{\infty}$ is said to be *realizable* if there is a solution of the problem, the system (A, B, C) is called *a realization* of the sequence $\{G_k\}_{k=1}^{\infty}$ and order n of the matrix A is *the order of the realization*. The sequence is realizable if and only if the infinite Hankel matrix

$$\begin{pmatrix} G_1 & G_2 & G_3 & \ldots \\ G_2 & G_3 & G_4 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{2}$$

has the finite rank ρ.

A realization is *minimal* if its order is minimal. The minimal realization is unique up to a similarity transformation. The order δ of the minimal realization coincides with the rank ρ. All existing algorithms of the minimal realization require the a priori knowledge of an upper bound δ_0 for the order of the realization: $\delta \leqslant \delta_0$. If such an estimate is known then for construction of the realization the finite sequence $G_1, G_2, \ldots, G_{2\delta_0}$ of the Markov parameters is required.

1.2. Construction of the Minimal Realization by the Transfer Function of a System

Given a $p \times q$ matrix function $G(z)$ (*the transfer function of a system*). Find a triplet $\Sigma = (A, B, C)$ of matrices such that

$$G(z) = C(zI - A)^{-1}B. \tag{3}$$

The matrix function $G(z)$ is *realizable* if there is a solution of the problem, the system (A, B, C) is called *a realization* of the matrix function $G(z)$, the order n of the matrix A is *the order of the realization*. The sequence is realizable if and only if $G(z)$ is a strictly

proper rational matrix function. A realization of $G(z)$ is *minimal* if its order is minimal. The order δ of the minimal realization of $G(z)$ is said to be *the McMillan degree* of the matrix function $G(z)$. There is an upper bound for the McMillan degree [5, Sec. 3.2]:

$$\delta \leqslant \delta_0 = \min\left\{\sum_{i=1}^{q}\alpha_i,\ \sum_{i=1}^{p}\beta_i\right\}. \tag{4}$$

Here α_i (β_i) is the degree of the lcd for entries of the i-th row (column) of the rational matrix function $G(z)$.

Let $G(z)$ be the realizable matrix function. Let us expand it as a Laurent series about $z=\infty$:

$$G(z) = \sum_{k=1}^{\infty}\frac{CA^{k-1}B}{z^k}.$$

Hence, the Laurent coefficients of $G(z)$ are the Markov parameters of the system. The matrix function $G(z) = \sum\limits_{n=1}^{\infty}\frac{G_k}{z^k}$ is a strictly proper rational matrix function if and only if the Hankel matrix (2) has a finite rank. Thus, realization Problem 1.1 and 1.2 are equivalent.

Informaly, the realization problem is to reconstruct a discrete-time linear time-invariant system Σ

$$\begin{cases} x_{j+1} &= Ax_j + Bu_j, \\ y_j &= Cx_j \end{cases},$$

that generates the given impulse response $\{G_k\}_{k=1}^{\infty}$. Note that the impulse response of the system can be measured and, in practice, the measured data can be disturbed by noise.

We identify the system Σ with the triple (A,B,C). The system $\Sigma_m = (A_m, B_m, C_m)$, that corresponds to the minimal realization, generates the same impulse response as Σ. However, Σ_m has important additional properties: it is observable and controllable.

A system (A,B,C) of order n is called *observable* if the observability matrix

$$\mathcal{O}_n(C,A) = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \tag{5}$$

has full rank. Similarly, the system is called *controllable* if the controllability matrix

$$\mathcal{C}_n(B,A) = \begin{pmatrix} B & AB & \cdots & A^{n-1}B \end{pmatrix} \tag{6}$$

has full rank.

By the Kalman theorem [6, Th. 6.2-3], a realization is minimal if and only if it is observable and controllable. The condition that $\mathcal{O}_n, \mathcal{C}_n$ have full rank is use as a minimality test.

1.3. Construction of the Minimal Realization by Reduction of a Non-Minimal Realization

Given a system $\Sigma = (A,B,C)$ of order n that is non-minimal. Find a minimal realization $\Sigma_m = (A_m, B_m, C_m)$ of the system.

Here we have the upper bound $\delta \leqslant \delta_0 = n$.

2. Algorithms and Procedures

In this section we describe algorithms for solving of these problems with the package MinimalRealization.

The package is designed as a user library and consists of 11 procedures: ApproxEssPoly, ApproxNullSpace, Approxrank, ExactEssPoly, FractionalFactorizationG, FractionalFactorizationMP, MarkovParameters, MinimalityTest, MinimalRealizationG, MinimalRealizationMP, Realization2MinimalRealization.

The main tools for solving of the realization problem are indices and essential polynomials of a finite matrix sequence [1, 2]. The procedures ExactEssPoly, ApproxEssPoly allow to find these characteristics of the sequence. ExactEssPoly is used for computation in rational arithmetic and ApproxEssPoly is applied if input data are floating-point numbers. In order to find the indices and the essential polynomials we should be able to calculate the ranks and null spaces of matrices. In the procedure ExactEssPoly we use the commands Rank and NullSpace of the LinearAlgebra package. In the presence of rounding errors finding of ranks or null spaces are ill-posed problems. The most widely used method for solving of these problems is the singular value decomposition (SVD). We also use SVD in the procedures Approxrank and ApproxNullSpace. To determinate the numerical ε-rank of a matrix it is required to choose the tolerance `epsilon`. This is a critical point of the problem and many works are devoted to this issue (see, e.g., [7]).

In the procedures Approxrank, ApproxNullSpace the tolerance `epsilon` is a natural number and the singular numbers of a matrix are treated as zero if they are less than $10^{-\text{epsilon}}$.

The indices and essential polynomials allow to construct a right fractional factorization of the transfer matrix of a system [1, Theorem 2], [8]. We can do this both by the matrix function $G(z)$ (the procedure FractionalFactorizationG) and by the sequence of the Markov parameters (the procedure FractionalFactorizationMP). We note that a construction of the fraction factorizations has an independent significance because these factorizations also play an important role in a problem of a generalized inversion of Toeplitz and Hankel matrices [9] and in a Wiener-Hopf factorization problem [8].

The minimal realization can be constructed if we know the fractional factorization [1, Theorem 3]. An implementation of the algorithm from this article carried out in the procedures MinimalRealizationMP, MinimalRealizationG and Realization2MinimalRealization. They solve Problems 1.1, 1.2, and 1.3, respectively. The procedure MinimalityTest is applied to test the minimality of the realization. For this we check that matrices (5) and (6) have the full ranks.

We now describe algorithms that are used in the procedures.

2.1. The Algorithm for Calculation of the Indices and Essential Polynomials. The Procedures ExactEssPoly, ApproxEssPoly

The parameters, that are passed to ExactEssPoly, are `M`, `N`, `cMN`. Here $M < N$ are integers, `cMN` is a sequence $c_M, c_{M+1}, \ldots, c_N$ of complex $p \times q$ matrices written in the form of the block row $\texttt{cMN} := \begin{pmatrix} c_M \\ c_{M+1} \\ \vdots \\ c_N \end{pmatrix}$.

For ApproxEssPoly we must also pass the tolerence `epsilon` that is used in Approxrank and ApproxNullSpace for the numerical determination of the ε-rank of a matrix.

We impose a technical restriction on the sequence `cMN`: $\omega = 0$ (the definition of the defect ω of the sequence see below in the description of Algorithm 1). We use the packages LinearAlgebra, PolynomialTools, numapprox of Maple.

The procedures return the indices $\mu_1, \dots, \mu_{p+q}$ and the matrix Rho $:= \big(R_1(z) \dots R_{p+q}(z)\big)$ consisting of right essential polynomials of the sequence `cMN`.

The algorithm of an exact calculation of μ_i и Rho is as follows (for details, see [1, 2]).

Algorithm 1. Exact calculation of indices and essential polynomials

procedure ExactEssPoly(M, N, cMN)
  Formation of the sequence $c_M, c_{M+1}, \dots, c_N$ by the matrix `cMN`;
  for k **from** `M` **to** `N` **do** $T_k = \|c_{i-j}\|_{\substack{i=k,k+1,\dots,N \\ j=0,1,\dots,k-M}}$;
  end do;
  $d_{M-1} := 0$, $d_{N+1} := (N - M + 2)q$;
  for k **from** `M` **to** `N` **do** rank:=**Rank**(T_k), $d_k := (k-M+1)q -$ rank, $\Delta_{k+1} := d_{k+1} - d_k$;
  end do;
  $\alpha := d_M$, $\omega := p + q - \Delta_{N+1}$;
  if $\omega \neq 0$ **then**
    Info: "error"; STOP;
  end if;
  Determination of the different indices $\tilde{\mu}_1, \dots, \tilde{\mu}_s$ and their multiplicities;
  Formation of the sequence $\mu_1, \dots, \mu_{p+q}$ of the indices taking into account their multiplicities;
  if $\sum_{i=1}^{p+q} \mu_i \neq (N+1)p + (M-1)q$ **then**
    Info: "error"; STOP;
  end if;
  for m **from** 1 **to** s **do** $\mathrm{kern_m} :=$ **NullSpace**$(\mathrm{T}_{\tilde{\mu}_m+1})$;
  end do;
  Formation of the matrix $\Lambda^{(\varkappa_1)}$ consisting of the first $\varkappa_1$ columns of the matrix Λ from the criterion of essentialness (see [2], Theorem 4.1);
  if **Rank**$(\Lambda^{(\varkappa_1)}) \neq \varkappa_1$ **then**
    Info: "error"; STOP;
  end if;
  $\Lambda := \Lambda^{(\varkappa_1)}$, Rho $:= \big(R_1(z) \dots R_{\varkappa_1}(z)\big)$, $R_j \in \mathrm{kern}_1$;
  for m **from** 2 **to** s **do**
    for k **from** 1 **to** $\dim \mathrm{kern}_m$ **do**
      Formation of the column Λ_k corresponging to vector $U_k \in \mathrm{kern}_{\varkappa_m}$;
      if **Rank**$(\Lambda, \Lambda_k) >$ **Rank**(Λ) **then** $\Lambda := (\Lambda, \Lambda_k)$, Rho $:= (\mathrm{Rho}, U_k(z))$;
      end if;
    end do;
  end do;
  if **Rank**$(\Lambda) \neq p + q$ **then**
    Info: "error"; STOP;
  end if;
end procedure

In the procedure ApproxEssPoly we use Approxrank and ApproxNullSpace instead of the functions Rank and NullSpace.

The following example shows that we have to use ApproxEssPoly in the presence of rounding errors or for noise input data.

Example 1. Let us consider the matrix sequence from Example 1 of the article [1]:

$$G_1 = \begin{pmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \end{pmatrix}, \; G_2 = \begin{pmatrix} 0 & 0 \\ 0 & -2 \\ 0 & 1 \end{pmatrix}, \; G_3 = \begin{pmatrix} 0 & -1 \\ 0 & 0 \\ 0 & 1 \end{pmatrix},$$

$$G_4 = \begin{pmatrix} 0 & 1 \\ 0 & -4 \\ 0 & 1 \end{pmatrix}, \; G_5 = \begin{pmatrix} 0 & -3 \\ 0 & 4 \\ 0 & 1 \end{pmatrix}, \; G_6 = \begin{pmatrix} 0 & 5 \\ 0 & -12 \\ 0 & 1 \end{pmatrix}.$$

We form the matrix cMN and call the procedure ExactEssPoly (hereinafter for brevity we will delete some lines from Maple worksheets):

```
> ExactEssPoly(1, 6, cMN);
                                    Indicies
 mu[1]=1, mu[2]=2, mu[3]=6, mu[4]=6, mu[5]=6,
                "alpha, omega, mu, Rho, L are now available"
> Rho;
            [[1,-1/2 z^2,0,z^6,0],[0,-1/2-1/2 z+z^2,z^6,0,z^5]]
```

The procedure returns the indices μ_j and the matrix Rho. Since the condition $\mu_5 - \mu_1 \leqslant 1$ is not satisfied, the indices of the sequence are unstable under small perturbations.

Let us generate a random perturbation of the matrix cMN by the Maple function RandomMatrix:

```
>RandG:=cMN+RandomMatrix(18,2,density=1,generator=-10^(-6)..10^(-6));
```

Now we call the procedure ExactEssPoly again. The following result is obtained:

```
> ExactEssPoly(1, 6, RandG);
                                    Indicies
 mu[1]=4, mu[2]=4, mu[3]=4, mu[4]=4, mu[5]=5,
                "alpha, omega, mu, Rho, L are now available"
```

Thus, the use of the exact algorithm does not allow to find the correct values of the indices.

Now we call the procedure ApproxEssPoly with the tolerance epsilon = 5:

```
> ApproxEssPoly(1, 6, RandG, 5);
                          "Indicies of the sequence:"
 mu[1]=1, mu[2]=2, mu[3]=6, mu[4]=6, mu[5]=6,
               "alpha, omega, mu, dmu, Rho are now available"
> Rho;
[[0.99999999999983568,
```

```
-0.18631934453218304+0.389874976941426200 z-0.340856515264002624 z^2,
0.,0.,1. z^6],
[0.,-0.340856623124891012-0.340856551772813099 z+0.681713292957929928 z^2,
1.,1. z,0.]]
```

We obtain for the perturbed sequence the same indices as for the unperturbed one. Note that we do not compare the matrices `Rho` consisting of right essential polynomials because a basis of the null space of a matrix is not uniquely determined.

2.2. The Algorithm for Construction of the Fractional Factorization. The Procedures FractionalFactorizationG, FractionalFactorizationMP

The procedure FractionalFactorizationG is applied for construction of a right fractional factorization

$$G(z) = N_R(z)D_R^{-1}(z)$$

of the transfer function $G(z)$ by Theorem 2 of the work [1]. For this we need the indices and the essential polynomials of the sequence $G_1, \ldots, G_m$, $m \geqslant 2\delta$, consisting of the Laurent coefficients of the rational matrix function $G(z)$.

The Markov parameters G_j are found by the procedure MarkovParameters. We pass to MarkovParameters two parameters: the matrix function `G` $:= G(z)$ and the number `mM` $:= m$ of the required parameters. To estimate the McMillan degree δ inequality (4) is applied. The procedure MarkovParameters returns the block column GMarkov consisting of the Markov parameters.

In order to find the indices and the essential polynomials we use the procedure ExactEssPoly if all entries of GMarkov are rational numbers, and we use ApproxEssPoly if there are floating-pointed numbers.

To the procedure FractionalFactorizationG we pass two parameters: `G` $:= G(z)$ and the tolerance `epsilon` that is used in ApproxEssPoly. The procedure returns the matrix polynomials DRight $:= D_R(z)$ and NRight $:= N_R(z)$.

The algorithm for the procedure is given bellow (see algorithm 2).

Algorithm 2. Construction of the fractional factorization

procedure FractionalFactorizationG(G, epsilon)
  Finding the upper bound δ_0 of the McMillan degree, mM $:= 2 * \delta_0$;
  GMarkov:=**MarkovParameters**(G,mM);
  if all entries of GMarkov are rational numbers **then**
    ExactEssPoly(1, mM, GMarkov);
  else
    ApproxEssPoly(1, mM, GMarkov, epsilon);
  end if;
  Selection of the submatrix $\big(R_1(z) \ldots R_q(z)\big)$ from Rho;
  DRight $:= \big(z^{\mu_1}R_1(z^{-1}) \ldots z^{\mu_q}R_q(z^{-1})\big)$;
  NRight1 $:= \sum_{1}^{\mathrm{mM}} G_j z^{-j} *$ DRight;
  NRight := the polynomial part of (NRight1);
end procedure

The procedure FractionalFactorizationMP is used to construct the right fractional factorization of $G(z)$ by the sequence $G_1, \ldots, G_m$, $m \geqslant 2\delta_0$, of the Markov parameters. The upper bound δ_0 of the McMillan degree δ must be known.

To the procedure FractionalFactorizationMP we pass three parameters: `mM` := m, `GMarkov`, `epsilon`. The procedure returns the matrices DRight := $D_R(z)$, NRight := $N_R(z)$. The algorithm for the FractionalFactorizationMP is the part of Algorithm 2 beginning with Line 4.

Example 2. Consider the matrix function

$$G(z) = \begin{pmatrix} \frac{1}{z} & -\frac{1}{z(z-1)(z+2)} \\ \frac{2}{z} & -\frac{2(z+1)}{z(z-1)(z+2)} \\ 0 & \frac{1}{z-1} \end{pmatrix}$$

from Example 1 of the article [1]. In this work we found the fractional factorization by the sequence of the Markov parameters. Now we construct the factorization directly by $G(z)$ with the help of the procedure FractionalFactorizationG.

```
> FractionalFactorizationG(G, 5);
                                    Indicies
 mu[1]=1, mu[2]=2, mu[3]=19, mu[4]=19, mu[5]=19,
                "alpha, omega, mu, Rho, L are now available"
                     DRight=[[z,-1/2],[0,-1/2 z^2-1/2 z+1]]
                     NRight=[[1,0],[2,1],[0,-1-1/2 z]]
```

We obtained the same result as in [1].

2.3. The Algorithms for Construction of the Minimal Realization. The Procedures MinimalRealizationMP, MinimalRealizationG and Realization2MinimalRealization

The minimal realization (A_m, B_m, C_m) of a system is found by Theorem 3 of the work [1]. For this we need the matrix polynomials $D_R(z)$, $N_R(z)$ from the right fractional factorization of the transfer function $G(z)$ of the system. Let us denote by D_{col} the invertible matrix consisting of leading coefficients of the columns of $D_R(z)$. In Theorem 3 [1] we showed that the matrices A_m, B_m, C_m can be constructed by entries of $D_{col}^{-1} D_R(z)$, D_{col}^{-1}, and $N_R(z)$, respectively.

We pass to the procedure MinimalRealizationMP (MinimalRealizationG) the same parameters as FractionalFactorizationMP (FractionalFactorizationG). To call Realization2MinimalRealization we must pass to it the matrices A, B, C of the non-minimal system Σ and the tolerence `epsilon`.

All procedures return the matrices Am := A_m, Bm := B_m, Cm := C_m that define the minimal realization of the system. The procedure MinimalityTest is used for checking minimality of the system. Algorithm 3 is the algorithm for MinimalRealizationMP.

The algorithm for MinimalRealizationG is the same as the previous one. The only difference is that we use the procedure FractionalFactorizationG instead of FractionalFactorizationMP.

Algorithm 3. Construction of the minimal realization by the Markov parameters

```
1: procedure MinimalRealizationMP(mM, GMarkov, epsilon)
2:     FractionalFactorizationMP(mM, GMarkov, epsilon);
3:     Finding of the matrix DRightCol := D_col;
4:     if ApproxRank(DRightCol) ≠ q then
5:         Info: "error"; STOP;
6:     end if;
7:     DRightNorm := (DRightCol)^{-1} * DRight;
8:     Formation of Am by DRightNorm;
9:     Formation of Bm by (DRightCol)^{-1};
10:    Formation of Cm by NRight;
11:    Test:=MinimalityTest(Am, Bm, Cm);
12:    if Test=0 then
13:        Info: "the system (Am, Bm, Cm) is minimal";
14:    else
15:        Info: "a minimal realization does not construct";
16:    end if;
17: end procedure
```

The algorithm for a reduction of a non-minimal realization to the minimal one has the following form (see Algorithm 4).

Algorithm 4. Reduction of a non-minimal realization to the minimal one

```
1: procedure Realization2MinimalRealization(A, B, C, epsilon)
2:     Test:=MinimalityTest(A, B, C);
3:     if Test=0 then
4:         Info: "the system (A, B, C) is minimal", STOP;
5:     else
6:         Info: "the system (A, B, C) is not minimal";
7:     end if;
8:     n:=RowDimension(A);
9:     for j from 1 to 2*n do G_j := CA^{j-1}B;
10:    end do;
11:    Formation of the matrix GMarkov;
12:    MinimalRealizationMP(2*n, GMarkov, epsilon);
13: end procedure
```

3. Numerical Experiments

Example 3. Let us find the minimal realization of the transfer function

$$G := \begin{pmatrix} z/(z+1)^2 & 1/z^2 \\ 1/(z+1) & 1/(z+2) \end{pmatrix}$$

from the article [5] with the help of MinimalRealizationG:

```
> MinimalRealizationG(G,4);
                                    Indicies
 mu[1]=2, mu[2]=3, mu[3]=11, mu[4]=12,
Am=[[0,1,0,0,0],[-1,-2,0,0,0],[0,0,0,1,0],[0,0,0,0,1],[0,0,0,0,-2]]
Bm=[[0,0],[1,0],[0,0],[0,0],[0,2]]
Cm=[[0,1,1,1/2,0],[1,1,0,0,1/2]]
```

Testing by formula (3) shows that (Am, Bm, Cm) is a realization of $G(z)$, the minimality test is also fulfilled.

Now we find the minimal realization for the perturbed matrix function

$$G1 := \begin{pmatrix} 0.9999999786z/(z+1.0000000361)^2 & 1.0/(z^2 \\ 1.0/(z+1.0000000361) & 0.999999941/(z+1.9999999234) \end{pmatrix},$$

with `epsilon`$=4$:

```
> MinimalRealizationG(G1,4);
                                    Indicies
 mu[1]=2, mu[2]=3, mu[3]=11, mu[4]=12,
Am=[[0,1,0,0,0],[-1.00000007699999993,-2.00000007700000015,-0.,-0.,-0.],
[0,0,0,1,0],[0,0,0,0,1],[-0.,-0.,-0.,-0.,-1.999999918999999996]]
Bm=[[0,0],[-2.44948983676636000,0.],[0,0],[0,0],[0.,2.23606790799958155]]
Cm=[[-1.90000000000000006 10^(-9),-0.408248266099999980,
0.894427182499999973, 0.447213609399999990,0.],
[-0.408248291300000010,-0.408248274799999978,2.99999999999999998 10^(-9),
-1.90000000000000006 10^(-9),0.447215835000000026]]
```

We can not compare the matrices Am, Bm, Cm for G и $G1$ because the minimal realization is unique up to a similarity transformation only. However, we see that the entries of the matrices Am for G и $G1$ have the same first 7 digits after decimal point. Moreover, using formula (3) we checked that we really had obtained an approximate minimal realization of G1. Hence, the algorithm is robust.

Example 4. To verify the procedure MinimalRealizationMP we consider the example from the work [10, pp. 433, 434]. Let us take the first 6 Markov parameters and put $x_1 = 17/3$, $x_2 = 1/37$. Since the input data are cumbersome, we do not cite them.

For the initial sequence the procedure returns the following values of indices

```
mu[1]=2, mu[2]=3, mu[3]=4, mu[4]=5,
```

Now we perturb the input data by RandomMatrix with the entries in the range $[-10^{-7}, 10^{-7}]$. For `epsilon` $= 6$ we get the following approximate minimal realization

```
mu[1]=2, mu[2]=3, mu[3]=4, mu[4]=5,
Am=[[0,1,0,0,0],[-2.00000856399999982,3.00000255500000002,
-4.95945338499999976,0.749432599000000032,-0.929948897000000052],
[0,0,0,1,0],[0,0,0,0,1],
[-0.0000157399999999999984,0.00000549999999999999986,-6.97297860000000026,
-3.33332246000000021,2.99999742000000014]]
```

```
Bm=[[0,0],[-15.3882633401427551,-20.6797652569520914],[0,0],[0,0],
[-41.9701472234983654,-41.9701418934536434]]
Cm=[[0.188982356099999998,-3.43999999999999970 10^(-8),
0.143975022300000005,-0.0931164741999999934,-0.0238264474300000016],
[-3.09695591100000002 10^(-8),-1.38333411999999998 10^(-9),
-0.0238265404300000012,1.84852747800000018 10^(-8),
-1.56090803199999994 10^(-9)]]
```

In order to verify the result we calculated the Markov parameters G_k for the initial sequence and the perturbed one. It turns out that the entries of the first 6 Markov parameters coincide up to the first 5 digits after decimal point.

Example 5. Here we test the procedure Realization2MinimalRealization. Let us consider the system (A, B, C), where

```
A := Matrix([[2/3, 0, -1, 3/7], [-2, 1, 1/2, 0], [0, 0, -2, 1/2],
[0, 0, 0, 0]]);
B := Matrix([[1/3, 0, -1], [-1, 1/2, 0], [-1/2, 0, 1], [0, 0, 0]]);
C := Matrix([[1, -3, -1, 1], [-1/2, 1, 1/2, -1/2]]);
```

We obtain the following result

```
> Realization2MinimalRealization(A, B, C, 5);
                      "The system (A,B,C) isn't minimal"
                        Am=[[9,0,-25/6],[49,1,-49/2],[22,0,-31/3]]
                        Bm=[[25/6,0,-9],[49/2,-1,-49],[31/3,0,-22]]
                        Cm=[[-11,3/2,5/4],[27/8,-1/2,-5/16]]
                      "The system (Am,Bm,Cm) is minimal"
```

Now we find the minimal realization if the initial data disturbed by noise.

```
A1 := A+RandomMatrix(4, 4, density = 1, generator = -10^(-6) .. 10^(-6));
B1 := B+RandomMatrix(4, 3, density = 1, generator = -10^(-6) .. 10^(-6));
C1 := C+RandomMatrix(2, 4, density = 1, generator = -10^(-6) .. 10^(-6));
Realization2MinimalRealization(A1, B1, C1, 5);
  Am=[[0.807667894000000052,-0.330770221899999994,0.0205231110000000002],
      [0.0257488389999999992,-1.16876715800000008,-0.975788304000000050],
      [0.230421724999999994,-1.75904463600000006,0.0277649500000000000]]
  Bm=[[5.62275499172367521,-0.876606563327936184,-9.13991031580920499],
      [10.8115998415014688,0.446659061140488090,-20.8163076602667552],
      [24.0189824469517284,-1.01594201125095229,-49.4999576454217234]]
  Cm=[[2.31639664900000009,0.156871677099999996,-0.453276107799999995],
      [-0.755875162299999958,-0.0473018757999999970,0.139258362200000014]]
```

For checking we calculated the first 10 Markov parameters for (A, B, C) and $(A1, B1, C1)$. The entries of these matrices coincide up to the first 4 digits after decimal point.

References

1. Adukov V.M. On a Problem of Minimal Realization. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming & Computer Software*, 2013, vol. 6, no. 3, pp. 5–17. (in Russian)
2. Adukov V.M. Generalized Inversion of Block Toeplitz Matrices. *Linear Algebra and Its Applications*, 1998, vol. 274, pp. 85–124. DOI: 10.1016/S0024-3795(97)00304-2
3. De Schutter B. Minimal State-Space Realization in Linear System Theory: An Overview. *Journal of Computational and Applied Mathematics, Special Issue on Numerical Analysis in the 20th Century. Vol. I: Approximation Theory*, 2000, vol. 121, no. 1-2, pp. 331–354.
4. Pushkov S.G., Krivoshapko S.Y. On the Problem of Realization in the State-Space Model for Interval Dynamic Systems. *Computing Technologies*, 2004, vol. 9, no. 1, pp. 75–85. (in Russian)
5. Sinha N.K. Minimal Realization of Transfer Function Matrices. A Comparative Study of Different Methods. *International Journal of Control*, 1975, vol. 22, no. 5, pp. 627–639. DOI: 10.1080/00207177508922109
6. Kailath T. *Linear Systems*. N.J., Prentice-Hall, Inc., Englewood Cliffs, 1980.
7. Foster L.V., Davis T.A. Algorithm 933: Reliable Calculation of Numerical Rank, Null Space Bases, Pseudoinverse Solutions, and Basic Solutions Using SuitesparseQR. *ACM Transactions on Mathematical Software (TOMS)*, 2013, vol. 40, issue 1, Article no. 7 (23 pages).
8. Adukov V.M. Fractional and Wiener-Hopf Factorizations. *Linear Algebra and Its Applications*, 2002, vol. 340, no. 1-3, pp. 199–213.
9. Adukov V.M. Generalized Inversion of Finite Rank Toeplitz and Hankel Operators with Rational Matrix Symbols. *Linear Algebra and Its Applications*, 1999, vol. 290, pp. 119–134. DOI: 10.1016/S0024-3795(98)10216-1
10. Tether A.J. Construction of Minimal Linear State-Variable Models from Finite Input-Output Data. *IEEE Transactions on Automatic Control*, 1970, vol. AC-15, no. 4, pp. 427–436. DOI: 10.1109/TAC.1970.1099514

Received August 13, 2014

УДК 519.71 **DOI: 10.14529/mmp140306**

РЕШЕНИЕ ЗАДАЧИ МИНИМАЛЬНОЙ РЕАЛИЗАЦИИ В СИСТЕМЕ MAPLE

В.М. Адуков, А.С. Фадеева

В системе компьютерной математики Maple создан пакет MinimalRealization для решения задачи минимальной реализации линейной конечномерной стационарной динамической системы с дискретным временем. Пакет позволяет построить минимальную реализацию системы по конечной последовательности марковских параметров

системы, либо по передаточной матрице-функции системы, либо по произвольной не минимальной реализации. Он оформлен в виде пользовательской библиотеки и состоит из 11 процедур: ApproxEssPoly, ApproxNullSpace, Approxrank, ExactEssPoly, FractionalFactorizationG, FractionalFactorizationMP, MarkovParameters, MinimalityTest, MinimalRealizationG, MinimalRealizationMP, Realization2MinimalRealization. Алгоритм реализации основан на последовательном решении трех задач: 1) нахождение индексов и существенных многочленов последовательности марковских параметров (процедуры ExactEssPoly, ApproxEssPoly), 2) построение правой дробной факторизации передаточной матрицы-функции (FractionalFactorizationG, FractionalFactorizationMP), 3) построение минимальной реализации по заданной дробной факторизации (MinimalRealizationG, MinimalRealizationMP, Realization2MinimalRealization). Предусмотрена возможность решения задачи как в условиях точных вычислений (в рациональной арифметике), так и при наличии ошибок округления или для начальных данных, возмущенных шумом. В последнем случае задача является неустойчивой, поскольку требует нахождения ранга и ядра матрицы. Используется сингулярное разложение матриц как наиболее надежный метод нахождения численного ранга (Approxrank) и ядра (ApproxNullSpace). Вычислительные эксперименты с пакетом MinimalRealization показали хорошее соответствие между точными и приближенными решениями задачи.

Ключевые слова: дискретная линейная конечномерная стационарная динамическая система; дробная факторизация; минимальная реализация; алгоритмы решения задачи реализации.

Литература

1. Адуков, В.М. О задаче минимальной реализации / В.М. Адуков // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. – 2013. – Т. 6, вып. 3. – С. 5–17.
2. Adukov, V.M. Generalized Inversion of Block Toeplitz Matrices / V.M. Adukov // Linear Algebra and Its Applications. – 1998. – V. 274. – P. 85–124.
3. De Schutter, B. Minimal State-Space Realization in Linear System Theory: An Overview / B.De Schutter // Journal of Computational and Applied Mathematics, Special Issue on Numerical Analysis in the 20th Century. Vol. I: Approximation Theory. – 2000. – V. 121, № 1-2. – P. 331–354.
4. Пушков, С.Г. О проблеме реализации в пространстве состояний для интервальных динамических систем / С.Г. Пушков, С.Ю. Кривошапко // Вычислительные технологии. – 2004. – Т. 9, № 1. – С. 75–85.
5. Sinha, N.K. Minimal Realization of Transfer Function Matrices. A Comparative Study of Different Methods / N.K. Sinha // International Journal of Control. – 1975. – V. 22, № 5. – P. 627–639.
6. Kailath, T. Linear Systems / T. Kailath. – N.J.: Prentice-Hall, Inc., Englewood Cliffs, 1980.
7. Foster, L.V. Algorithm 933: Reliable Calculation of Numerical Rank, Null Space Bases, Pseudoinverse Solutions, and Basic Solutions Using SuitesparseQR / L.V. Foster, T.A. Davis // ACM Transactions on Mathematical Software (TOMS). – 2013. – V. 40, issue 1. – Article No. 7 (23 pages).
8. Adukov, V.M. Fractional and Wiener-Hopf Factorizations / V.M. Adukov // Linear Algebra and Its Applications. – 2002. – V. 340, № 1-3. – P. 199–213.

9. Adukov, V.M. Generalized Inversion of Finite Rank Toeplitz and Hankel Operators with Rational Matrix Symbols / V.M. Adukov // Linear Algebra and Its Applications. – 1999. – V. 290. – P. 119–134.

10. Tether, A.J. Construction of Minimal Linear State-Variable Models from Finite Input-Output Data / A.J. Tether // IEEE Transactions on Automatic Control. – 1970. – V. AC-15, No. 4. – P. 427–436.

Виктор Михайлович Адуков, доктор физико-математических наук, профессор, кафедра «Математический и функциональный анализ», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), adukovvm@susu.ac.ru.

Анна Сергеевна Фадеева, ассистент, кафедра «Математический и функциональный анализ», Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), fadeevaas@susu.ac.ru.

Поступила в редакцию 13 августа 2014 г.