СБОРНИК **НАУЧНЫХ ТРУДОВ НГТУ. – 2018. – № 2 (92).** – 7–35

АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ И ИДЕНТИФИКАЦИЯ

УДК 681.513 DOI: 10.17212/2307-6879-2018-2-7-35

ПРЕОБРАЗОВАНИЕ ПОЛИНОМИАЛЬНОГО ПРЕДСТАВЛЕНИЯ МНОГОКАНАЛЬНОГО ОБЪЕКТА К ВЗАИМНО ПРОСТОМУ ВИДУ*

А.А. ВОЕВОДА[1], К.М. БОБОБЕКОВ[2]

[1] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор кафедры автоматики. E-mail: ucit@ucit.ru*

[2] *630087, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, аспирант кафедры автоматики. E-mail: kurbon_111@mail.ru*

Рассматривается приведение матричного полиномиального представления передаточной функции к взаимно простому виду. Нарушение требования взаимно простого представления приводит к появлению одинаковых корней в «числителе» и «знаменателе» как в одноканальных системах, так и в многоканальных. Это может привести к скрытой неустойчивости систем автоматического управления, неуправляемости или ненаблюдаемости; устранение подобных неприятностей иногда можно исключить введением дополнительных ограничений на задание корней желаемого характеристического полинома замкнутой системы. Кроме того, для существования решения полиномиального матричного уравнения при поиске регулятора требуется взаимно простое разложение передаточной функции объекта. Для проверки на взаимную простоту составляется система линейных однородных уравнений: другими словами, строится матрица Сильвестра и искомый матричный вектор из «элементов» взаимно простого представления. Далее при помощи команды $[q, r] = qr(S)$ пакета Matlab, где S – матрица Сильвестра, проверяются линейно зависимые столбцы в направлении слева направо. Если диагональные элементы матрицы r ненулевые, то исходные полиномиальные матрицы взаимно простые, а если присутствуют нули на диагонали – ищем правые взаимно простые полиномиальные матрицы. Приведены алгоритм вычисления взаимно простого представления и три примера, иллюстрирующие предлагаемый алгоритм.

Ключевые слова: матричная передаточная функция, левое и правое матричное полиномиальное разложение, взаимно простое разложение, не взаимно простое разложение, полиномиальные матрицы, линейное уравнение, матрица Сильвестра, вырожденная матрица, алгоритм поиска взаимной простоты

* Статья получена 10 января 2018 г.

ВВЕДЕНИЕ

Рассмотрим вычисление полиномиального представления передаточной функции к взаимно простому виду. Невыполнение требования взаимной простоты может привести к проблемам, связанным с появлением скрытой неустойчивости систем автоматического управления, неуправляемости или ненаблюдаемости [1–7]. Подобные затруднения могут быть исключены преобразованием полиномиального разложения к взаимно простому виду. Доказано, что для существования решения полиномиального матричного уравнения, которое возникает при поиске регулятора, гарантирующего заданный характеристический полином – в одноканальном случае, или заданную характеристическую полиномиальную матрицу – в многоканальном случае, требуется взаимно простое разложение. Для проверки на взаимную простоту удобно перейти от полиномиальных матричных уравнений к матричным уравнениям с числовыми коэффициентами [1, 2, 15].

Взаимно простое разложение может быть получено на основе алгоритма, суть которого состоит в поиске линейно зависимых столбцов матрицы Сильвестра слева направо. Для этого можно использовать *QR*-разложение (декомпозицию): в пакете Matlab соответствующий оператор *qr*. Предполагаем, и это действительно так, что у матрицы Сильвестра строк больше или равно количеству столбцов. Суть *QR*-разложения состоит следующем: для матрицы S размером $n \times m$ ($n \geq m$) существует ортогональная матрица $\bar{Q}$ размером $n \times n$ такая, что

$$\bar{Q}S = R ,$$

где R – верхнетреугольная матрица такого же размера, как S – $n \times m$. Так как $\bar{Q}$ «преобразует» строчки матрицы S, линейная независимость столбцов S «сохраняется» в столбцах R. Другими словами, если какой-либо столбец R линейно зависим от столбцов слева от него, таким же будет и столбец в S. Так как R – это верхнетреугольная матрица, ее m-й столбец линейно ***независим*** от столбцов с левой стороны, если и только если его m-й элемент на диагонали ***не нулевой***. Таким образом, анализируя диагональные элементы матрицы R слева направо, можем определить линейно зависимые/независимые столбцы матрицы S в зависимости от того, какие элементы стоят на диагонали, ноль или не ноль. Так как $\bar{Q}$ ортогональная, имеем $\bar{Q}^{-1} = \bar{Q}' = Q$ и $\bar{Q}S = R$, т. е. $S = QR$. Эта процедура называется *QR*-разложением (*decom-position*). В Matlab Q и R могут быть получены при помощи использования команды $[q, r] = qr(s)$.

Для скалярных передаточных функций можем использовать либо оператор *rank*, либо *QR*-разложение для определения количества μ линейно независимых *n*-столбцов. В матричном случае использование оператора *rank* неудобно и предлагается использовать *QR*-разложение.

Приведены алгоритм вычисления взаимно простого представления и примеры, подтверждающие работоспособность предлагаемого алгоритма. Данная работа является продолжением исследований, начатых в [7, 8], продолженных в [10, 14–17, 23] и опирающихся на базовые результаты, полученные в работах Chen.

1. ФОРМАЛИЗАЦИЯ ЗАДАЧИ ПРИВЕДЕНИЯ К ВЗАИМНО ПРОСТОМУ ВИДУ

Сделаем предположение, что матричная передаточная функция, например, объекта представлена в виде левого, возможно, не в виде взаимно простого полиномиального разложения[1]

$$W_{ob}(s) = D^{-1}(s)N(s)\,.$$

Поставим задачу поиска правого *взаимно простого разложения*:

$$W_{ob}(s) = \bar{N}(s)\bar{D}^{-1}(s)\,,$$

т. е. имеет место равенство

$$W_{ob}(s) = D^{-1}(s)N(s) = \bar{N}(s)\bar{D}^{-1}(s)\,.$$

[1] Можно сделать следующее пояснение о переходе от, например, правого не взаимно простого разложения $\bar{N}(s)\bar{D}^{-1}(s)$ к правому взаимно простому разложению. Допустим, каким-либо образом нашли наибольший общий правый делитель $R(s)$ матриц $\bar{N}(s)$ и $\bar{D}(s)$, т. е. $\bar{N}(s) = \hat{N}(s)R(s)$ и $\bar{D}(s) = \hat{D}(s)R(s)$. Тогда $\bar{N}(s)\bar{D}^{-1}(s) = \hat{N}(s)R(s)R^{-1}(s)\hat{D}^{-1}(s) = \hat{N}(s)\hat{D}^{-1}(s)$. Здесь $\hat{N}(s)$ и $\hat{D}(s)$ – правое взаимно простое разложение. Для дальнейших расчетов требуется, чтобы матрица $\hat{D}(s)$ была столбцово приведенной.

Очевидно, что справедливо

$$D(s)(-\bar{N}(s)) + N(s)\bar{D}(s) = 0 \, . \tag{1}$$

Матрицы с полиномиальными элементами, входящие в (1), можем записать в виде полиномов с матричными коэффициентами, у которых элементы – вещественные числа:

$$D(s) = D_0 + D_1 s + \cdots + D_{n-1}s^{n-1} + D_n s^n \, ,$$

$$N(s) = N_0 + N_1 s + \cdots + N_{m-1}s^{m-1} + N_m s^m \, ,$$

$$\bar{D}(s) = \bar{D}_0 + \bar{D}_1 s + \cdots + \bar{D}_{n-1}s^{n-1} \, ,$$

$$\bar{N}(s) = \bar{N}_0 + \bar{N}_1 s + \cdots + \bar{N}_{n-1}s^{n-1} \, .$$

Здесь D_i, N_i, $\bar{D}_i$ и $\bar{N}_i$ – числовые матрицы размером $p \times p$, p – число каналов и $m \le n$. Матрицы D_i и N_i нам известны, и необходимо всего лишь определить $\bar{D}_i$ и $\bar{N}_i$. Подставляя D_i, N_i, $\bar{D}_i$ и $\bar{N}_i$ в (1), выписываем систему линейных уравнений

$$\begin{array}{c} \begin{array}{ccccccc} D & N & D & N & \vdots & D & N \end{array} \\ \underbrace{\left[\begin{array}{c:c:c:c:c:c:c} D_0 & N_0 & 0 & 0 & \vdots & 0 & 0 \\ D_1 & N_1 & D_0 & N_0 & \vdots & 0 & 0 \\ \ldots & \ldots & D_1 & N_1 & \vdots & D_0 & N_0 \\ D_n & N_n & \ldots & \ldots & \vdots & D_1 & N_1 \\ 0 & 0 & D_n & N_n & \vdots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \vdots & D_n & N_n \end{array}\right]}_{S} \end{array} \cdot \underbrace{\left[\begin{array}{c} -\bar{N}_0 \\ \hdashline \bar{D}_0 \\ \hdashline -\bar{N}_1 \\ \hdashline \bar{D}_1 \\ \hdashline \ldots \\ \hdashline -\bar{N}_{n-1} \\ \hdashline \bar{D}_{n-1} \end{array}\right]}_{x} = 0 \, , \tag{2}$$

где S – матрица Сильвестра[2] размером $2n \times 2n$ блоков, столбцы которой обозначим через d_i и n_i: $D = (d_1, d_2, ..., d_p)$ и $N = (n_1, n_2, ..., n_p)$. Раскроем матрицу S и блочный вектор x:

[2] Транспонированные матрицы S тоже назовем матрицами Сильвестра.

$$\begin{array}{c} \begin{array}{ccccccccc} d_1 & d_2 & n_1 & n_2 & \vdots & d_1 & d_2 & n_2 & n_2 \end{array} \\ \left[\begin{array}{cc:cc:c:cc:cc} d_{11}^0 & d_{12}^0 & n_{11}^0 & n_{12}^0 & \vdots & 0 & 0 & 0 & 0 \\ d_{21}^0 & d_{22}^0 & n_{21}^0 & n_{22}^0 & \vdots & 0 & 0 & 0 & 0 \\ d_{11}^1 & d_{12}^1 & n_{11}^1 & n_{12}^1 & \vdots & d_{11}^0 & d_{12}^0 & n_{11}^0 & n_{12}^0 \\ d_{21}^1 & d_{22}^1 & n_{21}^1 & n_{22}^1 & \vdots & d_{21}^0 & d_{22}^0 & n_{21}^0 & n_{22}^0 \\ \cdots & \cdots & \cdots & \cdots & \vdots & d_{11}^1 & d_{12}^1 & n_{11}^1 & n_{12}^1 \\ d_{11}^n & d_{12}^n & n_{11}^n & n_{12}^n & \vdots & d_{21}^1 & d_{22}^1 & n_{21}^1 & n_{22}^1 \\ d_{21}^n & d_{22}^n & n_{21}^n & n_{22}^n & \vdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & d_{11}^n & d_{12}^n & n_{11}^n & n_{12}^n \\ 0 & 0 & 0 & 0 & \vdots & d_{21}^n & d_{22}^n & n_{21}^n & n_{22}^n \end{array}\right] \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \left[\begin{array}{cc} -\bar{n}_{11}^0 & -\bar{n}_{12}^0 \\ -\bar{n}_{21}^0 & -\bar{n}_{22}^0 \\ \hdashline \bar{d}_{11}^0 & \bar{d}_{12}^0 \\ \bar{d}_{21}^0 & \bar{d}_{22}^0 \\ \hdashline \cdots & \cdots \\ \hdashline -\bar{n}_{11}^{n-1} & -\bar{n}_{12}^{n-1} \\ -\bar{n}_{21}^1 & -\bar{n}_{22}^1 \\ \hdashline \bar{d}_{11}^{n-1} & \bar{d}_{12}^{n-1} \\ \bar{d}_{21}^{n-1} & \bar{d}_{22}^{n-1} \end{array}\right] \end{array} = 0 .$$

Для краткости записи матрица выписана для двухканальной системы $p = 2$.

Обсудим некоторые общие свойства матрицы S в предположении, что необходимо найти линейно независимые столбцы S слева направо. Это приводит нас к тому, что каждый D-столбец в каждом D-блочном столбце линейно независим от D-столбцов слева (из-за смещения вниз). Ситуация для N-столбцов, однако, отличная. Напомним, что здесь имеется p N-столбцов в каждом N-блочном столбце. Мы используем N_i -столбец для обозначения i-го N-столбца в каждом N-блочном столбце. Это приводит к тому, что если N_i -столбец в некотором N-блочном столбце линейно зависит от столбцов, входящих в N-блочные столбцы слева, тогда все последующие такие же N_i -столбцы ввиду повторяемости структуры S будут линейно зависимыми от столбцов слева. Пусть μ_i , $i = 1, 2, ..., p,$ будет число линейно независимых N_i -столбцов в S. Назовем их *столбцовыми индексами* (*column indices*) $W_{ob}(s)$. Первый N_i -столбец, линейно зависимый от столбцов слева, называют первым зависимым N_i -столбцом (*primary dependent* N_i *-column*). Ясно, что ($\mu_i + 1$)-й N_i -столбец – это первый зависимый столбец.

Формируем матрицу S_i , $i = 1, 2, ..., p$, которая включает в себя первый линейно зависимый столбец ($\mu_i + 1$), и вычисляем нормированный вектор x_i из ядра, i принимает значения $i = 1, 2, ..., p$.

Из таких нормированных векторов мы можем получить правое взаимно простое разложение. Результирующая матрица $D(s)$ имеет наименьшую возможную столбцовую степень. Дополнительно $\bar{D}(s)$ будет автоматически столбцово приведенной. Следующий пример иллюстрирует вышеприведенную процедуру.

2. ПРИМЕРЫ

Пример 1[3]. Найти взаимно простое правое полиномиальное матричное разложение передаточной функции

$$W_{ob}(s)=\begin{bmatrix} \dfrac{4s-10}{2s+1} & \dfrac{3}{s+2} \\ \dfrac{1}{(2s+1)(s+2)} & \dfrac{s+1}{(s+2)^2} \end{bmatrix}. \tag{3}$$

Сначала нужно найти левое разложение, не обязательно взаимно простое слева. Используя наименьший общий знаменатель каждой строки, можем легко получить

$$W_{ob}(s)=\begin{bmatrix} (2s+1)(s+2) & 0 \\ 0 & (2s+1)(s+2)^2 \end{bmatrix}^{-1}\times$$

$$\times\begin{bmatrix} (4s-10)(s+2) & 3(2s+1) \\ s+2 & (s+1)(2s+1) \end{bmatrix}=: D^{-1}(s)N(s).$$

Таким образом, имеем

$$D(s)=\begin{bmatrix} 2s^2+5s+2 & 0 \\ 0 & 2s^3+9s^2+12s+4 \end{bmatrix}=$$

$$=\underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}}_{D_0}+\underbrace{\begin{bmatrix} 5 & 0 \\ 0 & 12 \end{bmatrix}}_{D_1}s+\underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 9 \end{bmatrix}}_{D_2}s^2+\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}}_{D_3}s^3; \tag{4}$$

[3] Объект для анализа взять из Chen, который дан в виде матричной передаточной функции.

$$N(s)=\begin{bmatrix} 4s^2-2s-20 & 6s+3 \\ s+2 & 2s^2+3s+1 \end{bmatrix}=$$

$$=\underbrace{\begin{bmatrix} -20 & 3 \\ 2 & 1 \end{bmatrix}}_{N_0}+\underbrace{\begin{bmatrix} -2 & 6 \\ 1 & 3 \end{bmatrix}}_{N_1}s+\underbrace{\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}}_{N_2}s^2+\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_3}s^3. \qquad (5)$$

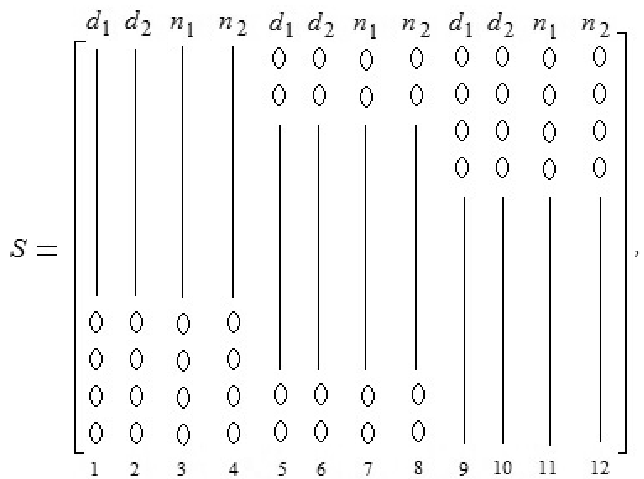
В нашем случае $n=3$ и каналов $p\times p=2\times 2$. Сформируем матрицу Сильвестра (2) размером $2n\times 2n=6\times 6$ блоков: 1-й и 2-й D- и N-блочные столбцы ($N=(n_1\ n_2\ \ldots\ n_p)$ и $D=(d_1\ d_2\ \ldots\ d_p)$), находящиеся рядом, дополняем снизу нулевыми матрицами для того, чтобы количество строк равнялось $2n$ блокам. Следующие блочные столбцы, соответствующие 3-му и 4-му столбцам, получим из первого и второго блочных столбцов со смещением вниз на одну позицию и дополняем сверху нулевыми матрицами. Аналогичным образом продолжим процедуру до того, что количество D- и N-блочных столбцов равнялось $2n$. Также формируем блочный столбец x из матричных коэффициентов полиномов $\bar{N}(s)$ и $\bar{D}(s)$ размером $2n\times 1$ блоков (2).

$$\begin{array}{cccccc} D & N & D & N & D & N \end{array}$$
$$\underbrace{\left[\begin{array}{c:c:c:c:c:c} D_0 & N_0 & O & O & O & O \\ D_1 & N_1 & D_0 & N_0 & O & O \\ D_2 & N_2 & D_1 & N_1 & D_0 & N_0 \\ D_3 & N_3 & D_2 & N_2 & D_1 & N_1 \\ O & O & D_3 & N_3 & D_2 & N_2 \\ O & O & 0 & 0 & D_3 & N_3 \end{array}\right]}_{S}\underbrace{\left[\begin{array}{c} -\bar{N}_0 \\ \hdashline \bar{D}_0 \\ \hdashline -\bar{N}_1 \\ \hdashline \bar{D}_1 \\ \hdashline -\bar{N}_2 \\ \hdashline \bar{D}_2 \end{array}\right]}_{x}=0,$$

где O – нулевая матрица размером $p\times p=2\times 2$, $\bar{N}_i$ и $\bar{D}_i$ – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов D_i и N_i из (4) и (5), то матрица Сильвестра принимает вид

$$S=\begin{array}{cc}
\begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} & \\
\left[\begin{array}{cc:cc:cc:cc:cc:cc}
2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 \\
0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 \\
0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 \\
0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 \\
0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0
\end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} \\
\begin{array}{cccccccccccc} d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \end{array} &
\end{array}$$

Матрицу S можно условно записать так:



где d_1, d_2, n_1 и n_2 – столбики размером восемь. Затем используем *QR*-разложение[4] для поиска линейно независимых столбцов в порядке слева направо в матрице S. Вычисления в пакете Matlab можно выполнить так:

```
>> d1=[2 0 5 0 2 0 0 0]; d2=[0 4 0 12 0 9 0 2]; % столбики размером восемь
   n1=[-20 2 -2 1 4 0 0 0]; n2=[3 1 6 3 0 2 0 0];
    S=[d1 0 0 0 0; d2 0 0 0 0; n1 0 0 0 0; n2 0 0 0 0;
     0 0 d1 0 0; 0 0 d2 0 0; 0 0 n1 0 0; 0 0 n2 0 0;
      0 0 0 0 d1; 0 0 0 0 d2; 0 0 0 0 n1; 0 0 0 0 n2]'; % транспонирование!
       [q, r]=qr(S); % Вычисление ортогональной матрицы q и матрицы
                      % r – верхнетреугольной таких, что q^-1 S = r
        r           % вывод матрицы r
```

Нам нужна только матрица *r*, а матрицу *q* не показываем. В результате получим матрицу *r*:

$$
\begin{array}{cc}
 & \begin{array}{cccccc} d1 & d2 & n1 & n2 & d1 & d2 \end{array} \\
r = & \left[\begin{array}{cccccc|}
-5,7 & 0 & 7,3 & -6,3 & -3,5 & 0 \\
0 & -15,7 & -1,3 & -3,7 & 0 & -11,1 \\
0 & 0 & -19,2 & 1,4 & -2,5 & 0,5 \\
0 & 0 & 0 & -2,0 & 3,4 & 2,9 \\
0 & 0 & 0 & 0 & -2,2 & 3,9 \\
0 & 0 & 0 & 0 & 0 & -9,9 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right. \\
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}
\end{array}
$$

[4] Для скалярных передаточных функций можем использовать либо оператор (команду) *rank*, либо команду *QR*-разложения для вычисления линейно зависимых столбцов (слева направо). В матричном случае использование команды *rank* неудобно и предлагается использование *QR*-разложение.

$$\begin{array}{c}
\begin{array}{cccccc} n1 & n2 & d1 & d2 & n1 & n2 \end{array} \\
\left|\begin{array}{cccccc}
18,1 & -4,7 & -0,7 & 0 & 7 & -1 \\
-2,1 & -2,8 & 0 & -3,8 & -1,3 & -1 \\
5,3 & -2,6 & -0,7 & 0,3 & 6,9 & -1 \\
6,6 & 4,5 & 1,7 & 3,2 & -16,4 & 3,3 \\
-4,6 & -0,5 & -4,8 & 4,7 & 4,2 & -4,6 \\
0,7 & -1,4 & -1,4 & -10,5 & -4,7 & -4,2 \\
-0,3 & 0,6 & -0,5 & -8,5 & -0,2 & -2,5 \\
0 & \boxed{0} & -0,5 & -3,2 & -4 & 0 \\
0 & 0 & -2 & 0,2 & 0 & 0 \\
0 & 0 & 0 & -2,5 & 0 & 0 \\
0 & 0 & 0 & 0 & \boxed{0} & 0 \\
0 & 0 & 0 & 0 & 0 & \boxed{0}
\end{array}\right] \\
\begin{array}{cccccc} 7 & 8 & 9 & 10 & 11 & 12 \end{array}
\end{array}.$$

Проанализируем линейную зависимость столбцов матрицы S при помощи матрицы r: видим, что все D-столбцы линейно независимые от столбцов слева, так как их элементы, соответствующие диагонали, ненулевые. Что касается N-столбцов, то всего два линейно независимых n_1-столбца (3-й и 7-й, $\mu_1 = 2$) и один линейно независимый n_2-столбец (4-й, $\mu_2 = 1$). Ищем первый линейно зависимый N-столбец слева направо – это соответствует n_2-столбцу (8-й, $\mu_2 + 1$), следующий линейно зависимый N-столбец слева направо – это n_1-столбец (11-й, $\mu_1 + 1$) и последний линейно зависимый N-столбец – это n_2-столбец (12-й).

Таким образом, из анализа вида матрицы r следует, что необходимо в матрице S оставить столбцы с 1-го по 8-й, и 8-й столбец – это $\mu_2 + 1$ – первый линейно зависимый столбец, и ввиду того, что две последние строки нулевые, их отбрасываем и обозначим через S_1. В результате получили матрицу

$$S_1 = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \left[\begin{array}{cc|cc|cc|cc} 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & -2 & 6 & 2 & 0 & -20 & 3 \\ 0 & 12 & 1 & 3 & 0 & 4 & 2 & 1 \\ 2 & 0 & 4 & 0 & 5 & 0 & -2 & 6 \\ 0 & 9 & 0 & 2 & 0 & 12 & 1 & 3 \\ 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 2 & 0 & 0 & 0 & 9 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \\ \begin{array}{cccccccc} d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \end{array} \end{array},$$

используя которую найдем вектор из ядра с помощью команды Matlab:

```
>> S1=[d1 0 0; d2 0 0; n1 0 0;, n2 0 0; 0 0 d1; 0 0 d2; 0 0 n1; 0 0 n2]';
   x2=null(S1);   % вектор из ядра
   x2b= x2/ x2(8)  % Нормирование вектора – 8-й элемент равен единице
```

В результате получим

$$x_{2b} = \begin{bmatrix} 7 & -1 & 1 & 2 & -4 & 0 & 2 & 1 \end{bmatrix}'.$$

Первый слева линейно зависимый 8-й столбец использовали. Следующий линейно зависимый столбец – 11-й, который соответствует $\mu_1 + 1$ – первый линейно зависимый n_1-столбец: матрицу S_2 формируем из столбцов слева направо по 11-й включительно, кроме 8-го:

$$S_2 = \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 9 & 10 & 11 \end{array} \\ \left[\begin{array}{cc|cc|cc|c|cc|c} 2 & 0 & -20 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & -2 & 6 & 2 & 0 & -20 & 0 & 0 & 0 \\ 0 & 12 & 1 & 3 & 0 & 4 & 2 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 5 & 0 & -2 & 2 & 0 & -20 \\ 0 & 9 & 0 & 2 & 0 & 12 & 1 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 4 & 5 & 0 & -2 \\ 0 & 2 & 0 & 0 & 0 & 9 & 0 & 0 & 12 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \end{array}\right] \\ \begin{array}{cccccccccc} d1 & d2 & n1 & n2 & d1 & d2 & n1 & d1 & d2 & n1 \end{array} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ \\ \end{array}.$$

Получили матрицу размером 12×10, у которой линейно зависимый 11-й столбец выделен другим цветом. Для определения вектора из ядра используем, как и выше, команды

```
>> d1=[2 0 5 0 2 0 0 0]; d2=[0 4 0 12 0 9 0 2];
     n1=[-20 2 -2 1 4 0 0 0]; n2=[3 1 6 3 0 2 0 0];
       S2=[d1 0 0 0 0; d2 0 0 0 0; n1 0 0 0 0;
         n2 0 0 0 0; 0 0 d1 0 0; 0 0 d2 0 0;
         0 0 n1 0 0;  0 0 0 0 d1; 0 0 0 0 d2;
       0 0 0 0 n1]';
     x1=null(S2);
   x1b= x1/ x1(10)
```

В результате получим

$$x_{1b} = \begin{bmatrix} 10 & -0,5 & 1 & 0 & 1 & 0 & 2,5 & -2 & 0 & 1 \end{bmatrix}'.$$

Таким образом, мы имеем

$$
\left[\begin{array}{c} -\bar{N}_0 \\ \hdashline \bar{D}_0 \\ \hdashline -\bar{N}_1 \\ \hdashline \bar{D}_1 \\ \hdashline -\bar{N}_2 \\ \hdashline \bar{D}_2 \end{array}\right] =
\left[\begin{array}{cc}
-\bar{n}_{11}^0 & -\bar{n}_{12}^0 \\
-\bar{n}_{21}^0 & -\bar{n}_{22}^0 \\ \hdashline
\bar{d}_{11}^0 & \bar{d}_{12}^0 \\
\bar{d}_{21}^0 & \bar{d}_{22}^0 \\ \hdashline
-\bar{n}_{11}^1 & -\bar{n}_{12}^1 \\
-\bar{n}_{21}^1 & -\bar{n}_{22}^1 \\ \hdashline
\bar{d}_{11}^1 & \bar{d}_{12}^1 \\
\bar{d}_{21}^1 & \bar{d}_{22}^1 \\ \hdashline
-\bar{n}_{11}^2 & -\bar{n}_{12}^2 \\
-\bar{n}_{21}^2 & -\bar{n}_{22}^2 \\ \hdashline
\bar{d}_{11}^2 & \bar{d}_{12}^2 \\
\bar{d}_{21}^2 & \bar{d}_{22}^2
\end{array}\right] =
\begin{array}{c}
\begin{array}{cc} x_{1b} & x_{2b} \end{array} \\
\left[\begin{array}{cc}
10 & 7 \\
-0,5 & -1 \\ \hdashline
1 & 1 \\
0 & 2 \\ \hdashline
1 & 4 \\
0 & 0 \\ \hdashline
2,5 & 2 \\
0 & 1 \\ \hdashline
-2 & 0 \\
0 & 0 \\ \hdashline
1 & 0 \\
0 & 0
\end{array}\right]
\end{array}.
$$

Можем выписать взаимно простые полиномиальные матрицы $\bar{D}(s)$ и $\bar{N}(s)$:

$$
\bar{D}(s) = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}}_{\bar{D}_0} + \underbrace{\begin{bmatrix} 2,5 & 2 \\ 0 & 1 \end{bmatrix}}_{\bar{D}_1} s + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}_{\bar{D}_2} s^2 = \begin{bmatrix} s^2 + 2,5s + 1 & 2s+1 \\ 0 & s+2 \end{bmatrix},
$$

$$
\bar{N}(s) = \underbrace{\begin{bmatrix} -10 & -7 \\ 0,5 & 1 \end{bmatrix}}_{\bar{N}_0} + \underbrace{\begin{bmatrix} -1 & 4 \\ 0 & 0 \end{bmatrix}}_{\bar{N}_1} s + \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}_{\bar{N}_2} s^2 = \begin{bmatrix} 2s^2 - s - 10 & 4s - 7 \\ 0,5 & 1 \end{bmatrix}.
$$

Таким образом, выпишем правое взаимно простое полиномиальное представление матрицы передаточной функции $\bar{D}(s)$ и $\bar{N}(s)$:

$$
W_{ob}(s) = \bar{N}(s)\bar{D}^{-1}(s) = \begin{bmatrix} (2s-5)(s+2) & 4s-7 \\ 0,5 & 1 \end{bmatrix} \times \begin{bmatrix} (s+2)(s+0,5) & 2s+1 \\ 0 & s+2 \end{bmatrix}^{-1}.
$$

Пример 2. Для двухканального динамического объекта в виде двух масс [10, 11], координаты которых y_1 и y_2 отсчитываются от состояния равновесия, и подвешенных последовательно на двух пружинах жесткости k_1 и k_2 решим задачу приведения описания к взаимно простому полиномиальному виду (для упрощения предполагаем отсутствие демпфирования). На каждую массу m_1 и m_2 действует управляемая внешняя сила u_1 и u_2. Найдем взаимно простые полиномиальные матрицы передаточной функции, которая дана в виде левого полиномиального разложения

$$W_{ob}(s) = D^{-1}(s)N(s) = \begin{bmatrix} 6s^2+3 & -2 \\ -2 & 2s^2+2 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

где

$$D(s) = \underbrace{\begin{bmatrix} 3 & -2 \\ -2 & 2 \end{bmatrix}}_{D_0} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D_1} s + \underbrace{\begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}}_{D_2} s^2, \tag{6}$$

$$N(s) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{N_0} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_1} s + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{N_2} s^2. \tag{7}$$

Сформируем матрицу Сильвестра (2)

$$\underbrace{\left(\begin{array}{c:c:c:c} D_0 & N_0 & O & O \\ D_1 & N_1 & D_0 & N_0 \\ D_2 & N_2 & D_1 & N_1 \\ O & O & D_2 & N_2 \end{array}\right)}_{S} \left(\begin{array}{c} -\bar{N}_0 \\ \hdashline \bar{D}_0 \\ \hdashline -\bar{N}_1 \\ \hdashline \bar{D}_1 \end{array}\right) = 0, \tag{8}$$

где O – нулевая матрица размером $p \times p = 2 \times 2$, $\bar{N}_i$ и $\bar{D}_i$ – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов D_i и N_i из (6) и (7) в (8) и раскрыть их, то получим

$$\underbrace{\begin{array}{c} \begin{array}{cccccccc} d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \end{array} \\ \left(\begin{array}{cc:cc:cc:cc} 3 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & 0 & 1 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{array}\right) \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \\ \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \end{array}}_{S} \underbrace{\left(\begin{array}{cc} -\bar{n}_{11}^{0} & -\bar{n}_{12}^{0} \\ -\bar{n}_{21}^{0} & -\bar{n}_{22}^{0} \\ \hdashline \bar{d}_{11}^{0} & \bar{d}_{12}^{0} \\ \bar{d}_{21}^{0} & \bar{d}_{22}^{0} \\ \hdashline -\bar{n}_{11}^{1} & -\bar{n}_{12}^{1} \\ -\bar{n}_{21}^{1} & -\bar{n}_{22}^{1} \\ \hdashline \bar{d}_{11}^{1} & \bar{d}_{12}^{1} \\ \bar{d}_{21}^{1} & \bar{d}_{22}^{1} \end{array}\right)}_{x} = 0,$$

где d_1, d_2, n_1 и n_2 – столбцы размером шесть. Процедура формулирования матрицы S и x делается аналогично, как в первом примере. Также для определения линейно независимых столбцов в порядке слева направо в матрице S используем QR-разложение:

```
>> d1=[3 -2 0 0 6 0]; d2=[-2 2 0 0 0 2]; % столбики размером шесть
   n1=[1 0 0 0 0 0]; n2=[0 1 0 0 0 0];
    S=[d1 0 0; d2 0 0; n1 0 0; n2 0 0;
   0 0 d1; 0 0 d2; 0 0 n1; 0 0 n2]'; % транспонирование!
  [q, r]=qr(S); % Вычисление ортогональной матрицы q и матрицы
                % r – верхнетреугольной таких, что q^-1 S = r
  r             % вывод матрицы r
```

В результате получим матрицу *r*:

$$r = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ d1 & d2 & n1 & n2 & d1 & d2 & n1 & n2 \end{array} \\ \begin{bmatrix} -7 & * & * & * & * & * & * & * \\ & -3,2 & * & * & * & * & * & * \\ & & -0,8 & * & * & * & * & * \\ & & & -0,7 & * & * & * & * \\ & & & & -7 & * & * & * \\ & & 0 & & & -0,3 & * & * \\ & & & & & & 0,8 & * \\ & & & & & & & -0,7 \end{bmatrix} \end{array}.$$

Все элементы d_i и n_i на диагонали – ненулевые, и это подтверждает, что все столбцы линейно не зависимые и полиномиальные матрицы $D(s)$ и $N(s)$ *взаимно простые*. Задача проверки взаимной простоты выполнена.

Пример 3. В качестве объекта возьмем камеру полимерной покраски, используемой в технологическом процессе покраски узлов и деталей на лифтостроительном производстве. Камера покраски, представляющая собой большой контейнер с движущимися подвешенными деталями, состоит из четырех соединенных между собой секций. Передаточная функция четырехканального объекта приведена в виде левого полиномиального разложения [8, с. 142]

$$W_{ob}(s) = D^{-1}(s)N(s) =$$

$$= \begin{bmatrix} s+0,1 & -0,015 & 0 & 0 \\ 0,01 & s+0,1 & -0,015 & 0 \\ 0 & 0,01 & s+0,1 & 0,01 \\ 0 & 0 & -0,015 & s+0,1 \end{bmatrix}^{-1} \begin{bmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0,1 \end{bmatrix}.$$

Выпишем матрицы $D(s)$ и $N(s)$ в виде полиномов:

$$D(s)=\underbrace{\begin{bmatrix}0,1 & -0,015 & 0 & 0\\ 0,01 & 0,1 & -0,015 & 0\\ 0 & 0,01 & 0,1 & 0,01\\ 0 & 0 & -0,015 & 0,1\end{bmatrix}}_{D_0}+\underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{D_1}s\,; \qquad (9)$$

$$N(s)=\underbrace{\begin{bmatrix}0,1 & 0 & 0 & 0\\ 0 & 0,1 & 0 & 0\\ 0 & 0 & 0,1 & 0\\ 0 & 0 & 0 & 0,1\end{bmatrix}}_{N_0}+\underbrace{\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}}_{N_1}s \qquad (10)$$

и проверим их на взаимную простоту, одновременно при этом вычисляется правое взаимно простое разложение. Но в данном случае этого можно не делать ввиду диагонального вида матрицы $N(s)$. Отметим, что полиномиальные матрицы N_i и D_i имеют размеры $p\times p=4\times 4$. Запишем систему линейных уравнений (2)

$$\underbrace{\begin{array}{c}\begin{array}{cc}D & N\end{array}\\ \begin{pmatrix}D_0 & \vdots & N_0\\ D_1 & \vdots & N_1\end{pmatrix}\end{array}}_{S}\underbrace{\begin{pmatrix}-\bar{N}_0\\ \hline \bar{D}_0\end{pmatrix}}_{x}=0\,, \qquad (11)$$

где $\bar{N}_i$ и $\bar{D}_i$ размером $p\times p=4\times 4$ – искомые взаимно простые полиномиальные матрицы. Если подставить значения элементов D_i и N_i из (9) и (10) в (11) и раскрыть, то (11) принимает следующий вид:

$$
\begin{array}{cccccccccc}
d1 & d2 & d3 & d4 & n1 & n2 & n3 & n4 & & \\
\left(\begin{array}{cccc|cccc}
0{,}1 & -0{,}015 & 0 & 0 & 0{,}1 & 0 & 0 & 0 \\
0{,}01 & 0{,}1 & -0{,}015 & 0 & 0 & 0{,}1 & 0 & 0 \\
0 & 0{,}01 & 0{,}1 & 0{,}01 & 0 & 0 & 0{,}1 & 0 \\
0 & 0 & -0{,}015 & 0{,}1 & 0 & 0 & 0 & 0{,}1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}\right)
& & & & & & & & \begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8 \end{array} &
\left[\begin{array}{cc|cc}
-\bar{n}_{11}^{0} & -\bar{n}_{12}^{0} & -\bar{n}_{13}^{0} & -\bar{n}_{14}^{0} \\
-\bar{n}_{21}^{0} & -\bar{n}_{22}^{0} & -\bar{n}_{23}^{0} & -\bar{n}_{24}^{0} \\
-\bar{n}_{31}^{0} & \bar{d}_{32}^{0} & -\bar{n}_{33}^{0} & \bar{d}_{34}^{0} \\
-\bar{n}_{42}^{0} & -\bar{n}_{42}^{0} & -\bar{n}_{43}^{0} & -\bar{n}_{44}^{0} \\
\hline
\bar{d}_{11}^{0} & \bar{d}_{12}^{0} & \bar{d}_{13}^{0} & \bar{d}_{14}^{0} \\
\bar{d}_{21}^{0} & \bar{d}_{22}^{0} & \bar{d}_{23}^{0} & \bar{d}_{24}^{0} \\
\bar{d}_{31}^{0} & \bar{d}_{32}^{0} & \bar{d}_{33}^{0} & \bar{d}_{34}^{0} \\
\bar{d}_{41}^{0} & \bar{d}_{42}^{0} & \bar{d}_{43}^{0} & \bar{d}_{44}^{0}
\end{array}\right] = 0. \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & &
\end{array}
$$

Здесь d_i, $i = 1, ..., 4$, и n_i, $i = 1, ..., 4$, – столбцы размером восемь. Как и в предыдущих примерах, ищем линейно независимые столбцы в матрице S в порядке слева направо:

```
>> d1=[0.1 0.01 0 0 1 0 0 0]; d2=[-0.015 0.1 0.01 0 0 1 0 0];
    d3=[0 -0.015 0.1 -0.015 0 0 1 0]; d4=[0 0 0.01 0.1 0 0 0 1];  % столбцы разме-
                                                                 % ром восемь
   n1=[0.1 0 0 0 0 0 0 0]; n2=[0 0.1 0 0 0 0 0 0];
   n3=[0 0 0.1 0 0 0 0 0]; n4=[0 0 0 0.1 0 0 0 0];
    S=[d1; d2; d3; d4; n1; n2; n3; n4]'; % транспонирование!
   [q, r]=qr(S); %
     r           % вывод матрицы r
```

В результате получим матрицу r:

$$
\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ d1 & d2 & d3 & d4 & n1 & n2 & n3 & n4 \end{array} \\
r = & \begin{bmatrix}
-1{,}005 & * & * & * & * & * & * & * \\
 & -1{,}005 & * & * & * & * & * & * \\
 & & -1{,}005 & * & * & * & * & * \\
 & & & -1{,}005 & * & * & * & * \\
 & & & & 0{,}1 & * & * & * \\
 & & 0 & & & 0{,}1 & * & * \\
 & & & & & & 0{,}1 & * \\
 & & & & & & & -0{,}1
\end{bmatrix}.
\end{array}
$$

Из вида матрицы r следует, что все элементы d_i и n_i на диагонали – ненулевые, что подтверждает линейную независимость столбцов. Делаем вывод, что полиномиальные матрицы $D(s)$ и $N(s)$ *взаимно простые*. Задача проверки взаимной простоты выполнена.

Приведем теорему, на основании которой разработаем алгоритм вычисления взаимно простого разложения матричной передаточной функции. Это необходимо сделать ввиду того, что *алгоритм* вычисления в [2] *не приведен*.

Теорема 1 [2, p. 219]. Пусть $W_{ob}(s) = D^{-1}(s)N(s)$ – левое матричное полиномиальное разложение, не обязательно взаимно простое. Используем коэффициентные матрицы $D(s)$ и $N(s)$ для формирования матрицы Сильвестра S и исследуем линейную независимость столбцов слева направо. Пусть μ_i, $i = 1, 2, \ldots, p$, будет число линейно независимых N_i-столбцов. Тогда

$$\deg W_{ob}(s) = \mu_1 + \mu_2 + \cdots + \mu_p ,$$

и правое взаимно простое разложение $\bar{N}(s)\bar{D}^{-1}(s)$ может быть получено вычислением p «штук» нормированных векторов из ядра p «штук» матриц, сформированных из каждого первого зависимого N_i-столбца и всех столбцов, линейно независимых слева от него.

3. АЛГОРИТМ ВЫЧИСЛЕНИЯ ВЗАИМНО ПРОСТОГО РАЗЛОЖЕНИЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ

Предполагаем, что задано левое матричное полиномиальное разложение (представление) передаточной функции

$$W_{ob}(s) = D^{-1}(s)N(s) ,$$

где

$$D(s) = \sum_{i=0}^{n} D_i s^i , \quad N(s) = \sum_{i=0}^{m} N_i s^i$$

такие, что $\deg N(s) \le \deg D(s)$, $m \le n$, $\dim D_i = \dim N_j = p \times p$, где p – число каналов. Отметим, что $n \ge 1$. В дальнейшем используем понятия «N-блочные столбцы» и «D-блочные столбцы»:

$$N = \left(N_0^t,\, N_1^t, \ldots,\, N_n^t\right)^t = (n_1\ \ n_2\ \ldots\ n_p)\,,\ D = \left(D_0^t,\, D_1^t, \ldots,\, D_n^t\right)^t = (d_1\ \ d_2\ \ldots\ d_p)\,.$$

Два блочных столбца D и N, стоящие рядом, дополняются нулевыми матрицами ($n \ge 2!$) сверху и снизу так, чтобы число (блочных) элементов в столбцах равнялось $2n$ блокам. В алгоритме осуществляется исследование уравнения (2), включающее в себя матрицу Сильвестра S и вектор матричных коэффициентов из взаимно простого представления матричной передаточной функции

$$x = (-\bar{N}_0^t \quad \bar{D}_0^t \mid -\bar{N}_1^t \quad \bar{D}_1^t \ \ \vdots\ \ -\bar{N}_{n-1}^t \quad \bar{D}_{n-1}^t)^t = (x_1\ \ x_2\ \ldots\ x_p)\,.$$

Предполагается использование пакета Matlab.

Алгоритм состоит из следующих действий.

1. Составление матрицы Сильвестра S размером $2n \times 2n$ блоков: первый и второй D- и N-блочные столбцы, стоящие рядом, дополняем снизу нулевыми матрицами для того, чтобы количество строк равнялось $2n$ блокам. Следующие блочные столбцы, соответствующие 3-му и 4-му столбцам, получим из 1-го и 2-го блочных столбцов смещением вниз на одну позицию и дополнением сверху нулевыми матрицами. Аналогичным образом строим $2n$ «штук» D- и N-блочных столбцов. Матрица Сильвестра сформирована. Формируем блочный столбец x из матричных коэффициентов полиномов $\bar{N}(s)$ и $\bar{D}(s)$ размером $2n \times 1$ блоков.

2. Вычисление линейно зависимых столбцов матрицы S в направлении слева направо, для чего используем команду $[q,\, r] = qr(S)$, которая выдает две матрицы – q и r; для дальнейших вычислений матрица q не требуется. Наличие линейно зависимых столбцов соответствует нулевым элементам на диагонали в верхнетреугольной матрице r.

3. Если все диагональные элементы матрицы r ненулевые, то полиномиальные матрицы $N(s)$ и $D(s)$ взаимно простые и, следовательно, ***выход из алгоритма***; если имеются нулевые элементы – ищем взаимно простые полиномиальные матрицы $\bar{N}(s)$ и $\bar{D}(s)$ – переход на следующий шаг.

4. Формирование матрицы S_1:

– так как все n «штук» D-блочных столбов матрицы S линейно независимы от всех столбцов слева, определяем количества $\mu_1, \mu_2, ..., \mu_p$ линейно независимых $n_1, n_2, ..., n_p$-столбцов слева направо в матрице S;

– матрица S_1 состоит из линейно независимых столбцов матрицы S, взятых слева направо, включая первый линейно зависимый n_1 ($\mu_1 + 1$-столбец), n_2 ($\mu_2 + 1$-столбец), …, n_p ($\mu_p + 1$-столбец) – остается первый линейно зависимый столбец;

– если в матрице S_1 имеется нулевая/нулевые строка/строки, то ее/их вычеркиваем – матрица S_1 сформирована.

5. Формируем вектор x_1, или x_2, или $\ldots x_p$: это зависит от того, какой столбец n_1, или n_2, или ..., n_p линейно зависимый оставили в матрице S_1: если n_1, то найдем x_1, а если n_2, то x_2 и т. д. У блочного вектора x_i вычеркиваем столько последних элементов, сколько столбцов вычеркнули при формировании матрицы S_1.

6. Для определения ненулевого решения x (состоит из матричных коэффициентов взаимно простых полиномов числителя и знаменателя матричной передаточной функции уравнения $S_1 x_1 = 0$, или $S_1 x_2 = 0$, или $\ldots S_1 x_p = 0$) воспользуемся командой Matlab ***null***(S_1) – вектор x_i принадлежит ядру S_1;

– найденный вектор нормируем – элементы вектора x_i делим на последний элемент его же и обозначим через x_{ib} (можно назвать нормированным вектором из ядра – *monic null vector*).

7. Формирование матрицы S_2, состоящей из линейно независимых столбцов матрицы S (предыдущий линейно зависимый столбец удаляется), взятых слева направо, включая первый линейно зависимый n_i-столбец (если в матрице S_1 n_2-столбец был линейно зависимый, то в матрице S_2 первый линейно зависимый столбец – n_1 или наоборот).

8. Действия шагов 4–7 выполняем для всех каналов, т. е. проводим вычисления для матриц $S_1, S_2, ..., S_p$.

9. Выпишем взаимно простые полиномиальные матрицы передаточной функции $\bar{N}(s)$ и $\bar{D}(s)$. Те столбцы, которые удалили из матриц

S_1, $S_2, \ldots, S_p$, соответствующие элементам векторов x_1, x_2, …, x_p, заполняем нулями.

Конец алгоритма.

ЗАКЛЮЧЕНИЕ

В теории автоматического управления используются различные виды описания объектов, например: в виде матричных передаточных функций $W(s)$, в пространстве состояний (A, B, C) и в виде матричных полиномиальных разложений $W(s) = D_l^{-1}(s)N_l(s) = N_r(s)D_r^{-1}(s)$ – левого и правого разложения [10, 14, 22, 23].

В работе предложен формализованный алгоритм поиска взаимно простого правого полиномиального матричного разложения $N(s)$ и $D(s)$ при использовании матрицы *Сильвестра* и команды в Matlab, базирующийся на теореме [Theorem 7.M4, p.219, 2]. Методики поиска взаимно простого разложения, рассмотренные в различных работах (например, в [1, 2], а также в диссертационных работах [7, 8]), приведены в недостаточно формализованном и четком виде, что вызывает определенные затруднения.

Задача взаимной простоты играет важную роль при решении задачи синтеза как в одноканальных системах [9, 16, 17, 25], так и в многоканальных системах [13, 18–20, 24]: взаимная простота полиномиальных матриц позволяет использовать так называемую матрицу Сильвестра или ее незначительные модификации при синтезе многоканальных регуляторов. Кроме того, если полиномы числителя и знаменателя передаточной функции объекта в одноканальных системах имеют одинаковые корни или полиномиальные матрицы не взаимно простые (имеют общий матричный полиномиальный не унимодальный множитель), это может привести к скрытой неустойчивости САУ, или неуправляемости / ненаблюдаемости системы. Если матрица Сильвестра не вырожденная, а это соответствует взаимной простоте полиномиального матричного разложения, то существует решение задачи синтеза многоканального регулятора, сводящееся к решению линейного алгебраического уравнения. С другими словами находят векторы из ядра матрицы Сильвестра. Для этого используется *QR*-разложение с целью поиска линейно зависимых столбцов.

На основе анализа нескольких примеров поиска взаимно простых правых полиномиальных матричных разложений $\bar{N}(s)$ и $\bar{D}(s)$ сформулирован *алгоритм вычисления взаимно простого разложения*.

СПИСОК ЛИТЕРАТУРЫ

1. *Kailath T.* Linear systems. – Englewood Cliffs, NJ: Prentice Hall, 1980. – 350 p.

2. *Chen C.T.* Linear system theory and design. – 3rd ed. – New York: Oxford University Press, 1999. – 334 p.

3. *Гайдук А.Р., Беляев В.Е., Пьявченко Т.А.* Теория автоматического управления в примерах и задачах с решениями в MATLAB: учебное пособие. – 2-е изд., испр. – СПб.: Лань, 2011. – 464 с.

4. *Ким Д.П.* Теория автоматического управления. Т. 2. Многомерные, нелинейные, оптимальные и адаптивные системы. – М.: Физматлит, 2004. – 464 с.

5. *Гайдук А.Р.* Теория и методы аналитического синтеза систем автоматического управления (полиномиальный подход). – М.: Физматлит, 2012. – 360 с.

6. *Гайдук А.Р.* Теория автоматического управления: учебник. – М.: Высшая школа, 2010. – 415 с.

7. *Вороной В.В.* Полиномиальный метод расчета многоканальных регуляторов пониженного порядка: дис. … канд. техн. наук: 05.13.01. – Новосибирск, 2013. – 173 с.

8. *Шоба Е.В.* Модальный метод синтеза многоканальных динамических систем с использованием полиномиального разложения: дис. … канд. техн. наук: 05.13.01. – Новосибирск, 2013. – 192 с.

9. *Воевода А.А., Ижицкая Е.А.* Стабилизация двухмассовой системы: модальный метод синтеза // Сборник научных трудов НГТУ. – 2009. – № 2 (56). – С. 3–10.

10. *Воевода А.А., Бобобеков К.М.* Решение переопределенной линейной системы уравнений при полиномиальном синтезе регуляторов // Современные технологии. Системный анализ. Моделирование. – 2017. – № 4 (56). – С. 84–99.

11. *Бобобеков К.М., Воевода А.А.* Синтез двухканальной системы полиномиальным методом: обеспечение астатизма // Сборник научных трудов НГТУ. – 2016. – № 1 (83). – С. 7–19.

12. *Воевода А.А.* Стабилизация двухмассовой системы: полиномиальный метод синтеза двухканальной системы // Сборник научных трудов НГТУ. – 2009. – № 4 (58). – С. 121–124.

13. *Воевода А.А., Вороной В.В*. Полиномиальный метод расчета многоканальных регуляторов заданной структуры // Научный вестник НГТУ. – 2013. – № 2 (51). – С. 214–218.

14. *Воевода А.А., Бобобеков К.М.* Синтез линейных многоканальных регуляторов с использованием структурных преобразований // Вестник Астраханского государственного технического университета. – 2017. – № 3. – С. 7–20.

15. *Бобобеков К.М., Тауров Э.Ш.* Вычисление взаимно простого разложения для одноканальных передаточных функций с использованием матрицы Сильвестра // Сборник научных трудов НГТУ. – 2018. – № 1 (91). – С. 7–30.

16. *Бобобеков К.М.* Полиномиальный метод синтеза одноканальной двухмассовой системы // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 25–36.

17. *Бобобеков К.М.* О нормировании полиномов знаменателей объекта и регулятора при полиномиальном методе синтеза // Сборник научных трудов НГТУ. – 2016. – № 4 (86). – С. 7–24.

18. *Воевода А.А*. Стабилизация двухмассовой системы: модальный метод синтеза с использованием полиномиального разложения // Научный вестник НГТУ. – 2010. – № 1 (38). – С. 195–198.

19. *Воевода А.А., Вороной В.В., Шоба Е.В.* Модальный синтез многоканального регулятора пониженного порядка с использованием «обратной» производной на примере трехмассовой системы // Научный вестник НГТУ. – 2012. – № 1 (46). – С. 15–22.

20. *Воевода А.А., Вороной В.В*. Модальный синтез регуляторов пониженного порядка методом дифференцирования характеристического полинома // Сборник научных трудов НГТУ. – 2011. – № 1 (63). – С. 3–12.

21. The modeling tests of the new PID-regulators structures / A.A. Voevoda, V.A. Zhmud, R.Y. Ishimtsev, V.M. Semibalamut // Proceedings of the 18th IASTED International Conference on Applied Simulation and Modelling, ASM 2009, 7–9 September, 2009, Palma de Mallorka, Spain. – [S. l.], 2009. – P. 165–168.

22. *Воевода А.А.* Матричные передаточные функции: (основные понятия): конспект лекций по курсу «Проектирование систем управления» для 4–5 курсов АВТФ (специальность 2101) / Новосибирский государственный технический университет. – Новосибирск: Изд-во НГТУ, 1994. – 94 с.

23. *Бобобеков К.М.* О структурных преобразованиях многоканальных линейных систем в матричном полиномиальном представлении // Научный вестник НГТУ. – 2017. – № 2 (67). – С. 7–25.

24. *Воевода А.А., Чехонадских А.В., Шоба Е.В.* Модальный метод синтеза с использованием полиномиального разложения: разделение движений при стабилизации трехмассовой системы // Научный вестник Новосибирского государственного технического университета. – 2011. – № 2 (43). – С. 39–46.

25. *Воевода А.А., Бобобеков К.М.* О необходимом условии существования решения при полиномиальном методе синтеза одноканальных систем // Сборник научных трудов НГТУ. – 2017. – № 4 (90). – С. 7–21.

Воевода Александр Александрович, доктор технических наук, профессор кафедры автоматики Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами. Имеет более 200 публикаций. E-mail: ucit@ucit.ru

Бобобеков Курбонмурод Мулломиракович, специалист по технологиям машиностроения, 2008–2013 гг. – кафедра «Технология машиностроения, металлорежущие станки и инструменты» механико-технологического факультета Таджикского технического университета (ТТУ) им. акад. М.С. Осими. С 2013 по 2015 г. ассистент Таджикского технического университета. С 2015 г. аспирант кафедры автоматики Новосибирского государственного технического университета. В настоящее время специализируется в области синтеза систем управления техническими системами. Имеет более 25 публикаций. E-mail: kurbon_111@mail.ru

DOI: 10.17212/2307-6879-2018-2-7-35

Transformation of a polynomial representation of a multichannel object to a coprime form*

A.A. Voevoda[1], K.M. Bobobekov[2]

[1] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, D. Sc. (Eng.), professor. E-mail: ucit@ucit.ru*

[2] *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk,630073, Russian Federation, the post-graduate student of Department Automatics. E-mail: kurbon_111@mail.ru*

Is considered bringing of the matrix polynomial representation of the transfer function of the object to a coprime form. Disarrangement requirements coprime representation leads to the appearance in the same roots "numerator" and "denominator" as in single-channel systems and in multi-channel systems. This can lead to latent instability of automatic control systems, unmanageability or non-observability; to eliminate such troubles, it can sometimes be ruled out by introducing additional restrictions on specifying the roots of the desired characteristic polynomial of the closed system. In addition, for the existence of a solution of a polynomial matrix equation in the search for a regulator, a mutually simple expansion of the transfer function of the object is required. To check for mutual simplicity, a system of linear homogeneous equations is compiled: in other words, the Sylvester matrix and sought-for matrix vector of the "elements" of the coprime representation are constructed. Then, using the $[q, r] = qr(S)$ command of the Matlab package, where S is the Sylvester matrix, the linearly dependent columns are checked in the direction from left to right. If the diagonal elements of r are non-zero, then the original polynomial matrices are coprime, and if there are zeros on the diagonal, we look for the right mutually simple polynomial matrices. An algorithm for calculating a coprime representation is presented and three examples illustrating the proposed algorithm are given.

* *Received 10 January 2018.*

Keywords: matrix transfer function, left and right matrix polynomial decomposition, coprime decomposition, the not coprime decomposition, polynomial matrices, linear equation, Sylvester matrix, the singular matrix, search algorithm coprime

REFERENCES

1. Kailath T. *Linear systems*. Englewood Cliffs, NJ, Prentice Hall, 1980. 350 p.

2. Chen C.T. *Linear system theory and design*. 3rd ed. New York, Oxford University Press, 1999. 334 p.

3. Gaiduk A.R., Belyaev V.E., P'yavchenko T.A. *Teoriya avtomaticheskogo upravleniya v primerakh i zadachakh s resheniyami v MATLAB* [Theory of automatic control in examples and problems with solutions in MATLAB]. 2nd ed., rev. St. Petersburg, Lan' Publ., 2011. 464 p.

4. Kim D.P. *Teoriya avtomaticheskogo upravleniya*. T. 2. *Mnogomernye, nelineinye, optimal'nye i adaptivnye sistemy* [The theory of automatic control. Vol. 2. Multidimensional, nonlinear, optimal and adaptive systems]. Moscow, Fizmatlit Publ., 2004. 464 p.

5. Gaiduk A.R. *Teoriya i metody analiticheskogo sinteza sistem avtomaticheskogo upravleniya (polinomial'nyi podkhod)* [Theory and methods of analytical synthesis of automatic control systems (polynomial approach)]. Moscow, Fizmatlit Publ., 2012. 360 p.

6. Gaiduk A.R. Teoriya avtomaticheskogo upravleniya [The theory of automatic control]. Moscow, Vysshaya shkola Publ., 2010. 415 p.

7. Voronoi V.V. *Polinomial'nyi metod rascheta mnogokanal'nykh regulyatorov ponizhennogo poryadka*. Diss. kand. tekhn. nauk [A polynomial method for calculating the multi-channel controllers low order. PhD eng. sci. diss.]. Novosibirsk, 2013. 173 p.

8. Shoba E.V. *Modal'nyi metod sinteza mnogokanal'nykh dinamicheskikh sistem s ispol'zovaniem polinomial'nogo razlozheniya*. Diss. kand. tekhn. nauk [The modal method for the synthesis of multi-channel dynamic systems using a polynomial expansion. PhD eng. sci. diss.]. Novosibirsk, 2013. 192 p.

9. Voevoda A.A., Izhitskaya E.A. Stabilizatsiya dvukhmassovoi sistemy: modal'nyi metod sinteza [Stabilization of two-mass systems: modal synthesis method]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2009, no. 2 (56), pp. 3–10.

10. Voevoda A.A., Bobobekov K.M. Reshenie pereopredelennoi lineinoi sistemy uravnenii pri polinomial'nom sinteze regulyatorov [Solution of an overdetermined linear system of equations for polynomial synthesis of regulators]. *Sov-*

remennye tekhnologii. Sistemnyi analiz. Modelirovanie – Modern Technologies. System analysis. Modeling, 2017, no. 4 (56), pp. 84–99.

11. Bobobekov K.M., Voevoda A.A. Sintez dvukhkanal'noi sistemy polinomial'nym metodom: obespechenie astatizma [Synthesis of two-channel system polynomial method: ensuring astatic]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 1 (83), pp. 7–19.

12. Voevoda A.A. Stabilizatsiya dvukhmassovoi sistemy: polinomial'nyi metod sinteza dvukhkanal'noi sistemy [Two-mass system stabilization: polynomial method of two-channel system synthesis]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2009, no. 4 (58), pp. 121–124.

13. Voevoda A.A., Voronoi V.V. Polinomial'nyi metod rascheta mnogokanal'nykh regulyatorov zadannoi struktury [Polynomial method for calculating multichannel controllers of a given structure]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2013, no. 2 (51), pp. 214–218.

14. Voevoda A.A., Bobobekov K.M. Sintez lineinykh mnogokanal'nykh regulyatorov s ispol'zovaniem strukturnykh preobrazovanii [Synthesis of linear multichannel regulators using structural transformations]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta – Vestnik of Astrakhan State Technical University*, 2017, no. 3, pp. 7–20.

15. Bobobekov K.M., Taurov E.Sh. Vychislenie vzaimno prostogo razlozheniya dlya odnokanal'nykh peredatochnykh funktsii s ispol'zovaniem matritsy Sil'vestra [Calculation of a mutually simple expansion for single-channel transfer functions using the Sylvester matrix]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 1 (91), pp. 7–30.

16. Bobobekov K.M Polinomial'nyi metod sinteza odnokanal'noi dvukhmassovoi sistemy [A polynomial method for the synthesis of single-channel two-mass system]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 25–36.

17. Bobobekov K.M. O normirovanii polinomov znamenatelei ob"ekta i regulyatora pri polinomial'nom metode sinteza [About rationing polynomials denominator object and regulator during polynomial method of synthesis]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2016, no. 4 (86), pp. 7–24.

18. Voevoda A.A. Stabilizatsiya dvukhmassovoi sistemy: modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya [Stabilisation of two-mass system by a modal method of synthesis with polynomial factorization]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2010, no. 1 (38), pp. 195–198.

19. Voevoda A.A., Voronoy V.V., Shoba E.B. Modal'nyi sintez mnogokanal'nogo regulyatora ponizhennogo poryadka s ispol'zovaniem "obratnoi" proizvodnoi na primere trekhmassovoi sistemy [Modal synthesis of multi-channel low-order controller using the "reverse" derivative principle for three-mass system]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2012, no. 1 (46), pp. 15–22.

20. Voevoda A.A., Voronoy V.V. Modal'nyi sintez regulyatorov ponizhennogo poryadka metodom differentsirovaniya kharakteristicheskogo polinoma [Modal design of reduced order controllers by method of differentiation of the characteristic polynomial]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2011, no. 1 (63), pp. 3–12.

21. Voevoda A.A., Zhmud V.A., Ishimtsev R.Y., Semibalamut V.M. The modeling tests of the new PID-regulators structures. *Proceedings of the IASTED International Conference on Applied Simulation and Modelling, ASM 2009*, Palma de Mallorka, Spain, 7–9 September, 2009, pp. 165–168.

22. Voevoda A.A. *Matrichnye peredatochnye funktsii: (osnovnye ponyatiya)* [Matrix transfer functions (basic concepts)]. Novosibirsk State Technical University. Novosibirsk, NSTU Publ., 1994. 94 p.

23. Bobobekov K.M. O strukturnykh preobrazovaniyakh mnogokanal'nykh lineinykh sistem v matrichnom polinomial'nom predstavlenii [About structural transformations of multichannel linear systems in the matrix polynomial representation]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2017, no. 2 (67), pp. 7–25.

24. Voevoda A.A., Chekhonadskikh A.V., Shoba E.V. Modal'nyi metod sinteza s ispol'zovaniem polinomial'nogo razlozheniya: razdelenie dvizhenii pri stabilizatsii trekhmassovoi sistemy [Modal synthesis method using a polynomial decomposition: the separation of motions in the stabilization of the three-mass plant]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2011, no. 2 (43), pp. 39–46.

25. Voevoda A.A., Bobobekov K.M. O neobkhodimom uslovii sushchestvovaniya resheniya pri polinomial'nom metode sinteza odnokanal'nykh sistem [About the necessary conditions of existence of the solution in polynomial method of syn-

thesis of single-channel systems]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2017, no. 4 (90), pp. 7–21.

Для цитирования:

Воевода А.А., Бобобеков К.М. Приведение матричного полиномиального представления передаточной функции к взаимно простому виду с использованием матрицы Сильвестра // Сборник научных трудов НГТУ. – 2018. – № 2 (92). – С. 7–35. – doi: 10.17212/2307-6879-2018-2-7-35.

For citation:

Voevoda A.A., Bobobekov K.M. Privedenie matrichnogo polinomial'nogo predstavleniya peredatochnoi funktsii k vzaimno prostomu vidu s ispol'zovaniem matritsy Sil'vestra [Transformation of a polynomial representation of a multichannel object to a coprime form]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2018, no. 2 (92), pp. 7–35. doi: 10.17212/2307-6879-2018-2-7-35.