RIGHT COPRIME FACTORIZATIONS FOR SINGLE-INPUT DESCRIPTOR LINEAR SYSTEMS: A SIMPLE NUMERICALLY STABLE ALGORITHM

Guang-Ren Duan

ABSTRACT

Based on the upper-triangular Hessenberg forms for descriptor linear systems and a method for right coprime factorization of descriptor linear systems using matrix elementary transformations, a simple, neat and numerically stable iterative formula for right coprime factorization of a regular single-input descriptor linear system is proposed. This iterative formula gives directly the coefficients of the solutions to the coprime factorization of the system, and involves only multiplication and additions of non-negative scalar terms. Numerical examples are presented to demonstrate the proposed algorithm.

KeyWords: Descriptor linear systems, right coprime factorization, upper-triangular Hessenberg forms, iterative solution, orthogonal transformations, numerical stability.

I. INTRODUCTION

Consider the following descriptor linear system:

\[ E \dot{x} = Ax + bu, \] (1)

where \( x \in \mathbb{R}^n \) is the state vector, \( u \in \mathbb{R} \) is the scalar input, and \( E, A \) and \( b \) are known real matrices of appropriate dimensions. The matrix \( E \) in (1) may be singular. In this case, the matrix pencil \( \det(sE - A) \) has an order less than the system dimension. The system (1) is called regular if the matrix pencil \( \det(sE - A) \) is not identically zero. Regularity is a very important feature for descriptor linear systems since it ensures the uniqueness of the solution of the system.

In the special case of \( E = I \), (1) represents a conventional linear system. When the matrix \( E \) is nonsingular, (1) can be converted into the following conventional one:

\[ \dot{x} = E^{-1}Ax + E^{-1}bu. \] (2)

Many practical systems are originally in the form of (1). For such systems with \( \det(E) \neq 0 \), techniques developed for conventional linear systems can indeed be adopted for analysis and design since they can be converted into the form of (2). However, such a conversion is generally not numerically reliable and often results in a conventional linear system with poor conditioning [1].

This paper is concerned with the following right coprime factorization of system (1):

\[ (sE - A)^{-1}b = N(s)d^{-1}(s), \] (3)

where \( N(s) \) is a polynomial vector of dimension \( n \), \( d(s) \) is a scalar polynomial and they are right coprime. In order that the equation (3) makes sense, the system (1) is required to be regular.

The above problem has a close relation with transfer function computation of descriptor linear systems in state-space representations. On one hand, when the above factorization problem is solved, the transfer function of the system (1) with the output equation \( y = Cx \) is obviously obtained as \( CN(s)d^{-1}(s) \). On the other hand, (3) can be viewed as the transfer function of system (1) with the special output equation \( y = x \). Therefore, approaches for computation of the transfer function from a descriptor linear system in state-space form, such as that in [2], can be readily applied to solve the above right coprime factorization problem. However, the algorithms proposed in [3, 4] and [5] suit only for the case of \( \det(E) \neq 0 \) and can not be applied to the descriptor linear system (1) when the matrix \( E \) is singular. The problem of computing the
coprime factorization of a given transfer function matrix has been widely studied (see, e.g., [6-9]). Theoretically speaking, these methods may be adopted to solve the factorization (3) by first obtaining the rational fraction form of \((sE - A)^{-1} b\). However, it is generally not numerically favorable to obtain the rational fraction form of \((sE - A)^{-1} b\) since it involves solving the inverse of \((sE - A)\).

Right coprime factorization of linear systems has many applications in control system theory. Green [10] has developed a coprime factorization approach to the synthesis of H-infinity controllers, and Armstrong [11] has considered robust stabilization using a coprime factorization approach, while Ohishi et al. [12] proposed a new speed servo system for a wide speed range based on doubly coprime factorization. Furthermore, it has been shown by the author and his co-authors that a coprime factorization can be used to parameterize all the solutions to a generalized-type of Sylvester matrix equations [13,14,15], and the solutions to the problems of eigenstructure assignment [13,14,16,17], robust pole assignment [18,19,20] and observer-based robust fault detection [21].

A direct solution to the right coprime factorization (3) can be easily seen to be

\[ N(s) = \text{adj}(sE - A)b, \quad D(s) = \det(sE - A), \quad (4) \]

but this solution is obviously not easy to compute and is generally not numerically reliable. In solving eigenstructure assignment in descriptor linear systems, Duan [13] has given a simple method for calculating a right coprime factorization for a linear descriptor system using matrix elementary transformations. This method is very efficient for lower order systems since accurate solutions can often be obtained easily by hand, but it is not convenient to use when the system dimension is relatively large.

Linear systems in state-space forms can be transformed, by using orthogonal algebraic transformations, into the system Hessenberg-triangular forms [1,2,22,23]. Since only orthogonal matrices or orthogonal similarity transformations are involved, Hessenberg form reduction is a numerically stable procedure [1]. Further, due to its special structure, system Hessenberg form offers a great convenience for a certain type of analysis and design problems (see, e.g., [1] and [22-27]). Based on the system lower Hessenberg form and a simple method proposed in [14] for right coprime factorization of conventional linear systems using matrix elementary transformations, the author has considered solutions to the right coprime factorizations for both single-input conventional linear systems [28,29,30] and multi-input conventional linear systems [31], and has established simple iterative formulas in polynomial or scalar format for solutions to the right coprime factorizations. Also, based on system triangular Hessenberg forms, Misra [2] has presented an algorithm for computation of the transfer function of a descriptor linear system in state-space form. However, special care has to be taken when using this algorithm because it contains division manipulations which may lead to floating point overflows or underflows if the sub-diagonal elements of the transformed Hessenberg form matrix are too small or too large.

The purpose of this paper is to derive a numerically stable algorithm for solving right coprime factorizations of regular single-input descriptor linear systems based on System Hessenberg form reduction. The development is based on the method proposed in [13] for right coprime factorizations of descriptor linear systems using matrix elementary transformations. By applying this method to the converted system Hessenberg form, a neat and simple iterative formula for right coprime factorization of a regular single-input descriptor system is derived, which gives directly the coefficients of the solutions to the right coprime factorization of the descriptor linear system. After modification, it involves only multiplication and addition manipulations of non-negative scalar terms.

This paper is divided into six sections. In the next section some definitions related to the system Hessenberg form of a single-input descriptor linear system are given together with some preliminary results. In Section 3, right coprime factorization of a single-input descriptor linear system based on upper-triangular Hessenberg form is considered. Numerical examples are worked out in Section 4 to demonstrate the effect of the proposed approach. The concluding remark is given in Section 5. An appendix which provides the proof of the Theorem 3.1 is provided following the references.

II. SYSTEM HESSENNBERG FORMS

For convenience, a descriptor linear system in the form of (1) will also be referred as \((E A b)\) in the following.

Definition 2.1. A single-input descriptor system \((H^v_H^u_H^v_H^u)\) of order \(v\) is said to be in generalized upper-triangular Hessenberg form (GUHF) if

\[ H^v_H^u = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1,v-1} & h_{1,v} \\ h_1 & h_{22} & \cdots & h_{2,v-1} & h_{2,v} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & h_{v-1,v} & h_{v,v} \end{bmatrix}, \quad (5a) \]

and

\[ H^v_H^u = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1,v} \\ 0 & e_{22} & \cdots & e_{2,v} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & e_{v,v} \end{bmatrix}, \quad H^v_H^u = \begin{bmatrix} h_v & \cdots & 0 \\ 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots \\ 0 & \cdots & 0 \end{bmatrix}. \quad (5b) \]
where \( h_0 \neq 0 \).

It has been shown in [1,22] and [24] that, for an arbitrary single-input descriptor system \((E A b)\) of order \( v\), there always exist a pair of orthogonal matrices \( T \) and \( S \) such that \((SET SAT Sb)\) is in GUHF. An efficient and numerically stable algorithm for transforming a single-input descriptor system into a GUHF can also be found in [1,22] and [24].

**Definition 2.2.** Let \((S T)\) be a pair of orthogonal matrices which transform the \(\nu\)-th order single-input descriptor linear system \((E A b)\) into a GUHF \((H^n_E H^n_A H^n_b)\). Then, \((H^n_E H^n_A H^n_b)\) is called a GUHF for \((E A b)\) associated with \((S T)\).

The descriptor linear system \((E A b)\) of order \( v\) is said to be controllable if

\[
\text{rank}[sE - A \ b] = v, \quad \forall s \in C.
\]

Obviously, the GUHF \((H^n_E H^n_A H^n_b)\) is controllable if \(h_i \neq 0, \ i = 1, 2, \ldots, v - 1\). Define

\[
n = \begin{cases} 
  i, & \text{if } h_i \neq 0, \ h_i^v = 0, \ 1 \leq j < i \leq v - 1 \\
  v, & \text{if } h_i \neq 0, \ i = 1, 2, \ldots, v - 1 
\end{cases}
\]

(6)

Then, according to the structure theory of descriptor linear systems [32], it is clear that the \(n\)-th order system \((H^n_E H^n_A H^n_b)\) is the controllable subsystem of \((H^n_E H^n_A H^n_b)\). Based on this fact, we can further introduce the following concept.

**Definition 2.3.** Let \((H^n_E H^n_A H^n_b)\) be a GUHF for \((E A b)\) associated with \((S T)\), and \(n\) be the integer defined by (6). Then, the \(n\)-th order system \((H^n_E H^n_A H^n_b)\) is called the GUHF controllable subsystem of \((E A b)\) associated with \((S T)\).

The following lemma gives the relation between the right coprime factorization of a single-input descriptor linear system and that of its GUHF controllable subsystem.

**Lemma 2.1.** Let \((H^n_E H^n_A H^n_b)\) be the GUHF controllable subsystem of the \(\nu\)-th order single-input descriptor linear system \((E A b)\) associated with \((S T)\). Denote by \(N_n(s)\) and \(d_n(s)\) a pair of solutions to the right coprime factorization of \((H^n_E H^n_A H^n_b)\). Then, a pair of solution \(N(s)\) and \(d(s)\) to the right coprime factorization of the descriptor system \((E A b)\) is given by

\[
N(s) = T_i N_n(s), \quad d(s) = d_n(s),
\]

(7)

where \(T_i \in R^{n \times n}\) is the matrix formed by the first \(n\) columns of the matrix \(T\).

**Proof.** Let \((H^n_E H^n_A H^n_b)\) be the GUHF of the \(\nu\)-th order single-input descriptor linear system \((E A b)\) associated with \((S T)\). Then it is easy to show the following:

\[
(sE - A)^{-1}b = T(sH^n_E - H^n_A)^{-1}H^n_b. \quad (8)
\]

Note that the right coprime factorization for \((H^n_E H^n_A H^n_b)\) can be viewed as the transfer function of the system \(\Sigma = (H^n_E H^n_A H^n_b I_b)\), which is a system with the output distribution matrix being the identity matrix \(I_b\). Since \((H^n_E H^n_A H^n_b)\) is the controllable subsystem of \((H^n_E H^n_A H^n_b)\), the following system

\[
\Sigma_n = (H^n_E H^n_A H^n_b \begin{bmatrix} I_n \\ 0_{n \times n} \end{bmatrix})
\]

is the strongly controllable subsystem of \(\Sigma\) [32]. Further note that the remaining \((\nu - n)\)-th order subsystem in \(\Sigma\) is uncontrollable and will lead to pole-zero cancellation in the transfer function of the system [2], the transfer function of the system \(\Sigma\) is thus equal to that of its strongly controllable subsystem \(\Sigma_n\). This leads the following relation:

\[
(sH^n_E - H^n_A)^{-1}H^n_b = I_n \begin{bmatrix} I_n \\ 0_{n \times n} \end{bmatrix}(sH^n_E - H^n_A)^{-1}H^n_b. \quad (9)
\]

Further note that \(N_n(s)\) and \(d_n(s)\) are a pair of solutions to the right coprime factorization of \((H^n_E H^n_A H^n_b)\), we thus have

\[
(sH^n_E - H^n_A)^{-1}H^n_b = N_n(s) d_n^{-1}(s). \quad (10)
\]

Combining (8)-(10), gives

\[
(sE - A)^{-1}b = T(sH^n_E - H^n_A)^{-1}H^n_b
\]

\[
= T_i (sH^n_E - H^n_A)^{-1}H^n_b
\]

\[
= T_i N_n(s) d_n^{-1}(s).
\]

This shows that the relations in (7) hold.

**III. MAIN RESULTS**

It follows from Lemma 2.1 that a right coprime factorization for a \(\nu\)-th order regular descriptor system \((E A b)\) can be easily obtained when a right coprime factorization for its GUHF controllable subsystem \((H^n_E H^n_A H^n_b)\) is available. Thus, let us first investigate the right coprime factorization of the GUHF controllable
subsystem \((H^a_k H^b_k H^c_k)\). For technical convenience, we re-
define the matrices \(H^a_k, H^b_k\) and \(H^c_k\) and their entries as follows:

\[
H^a_k = H^b_k = \begin{bmatrix}
h_{n,n} & h_{n,n-1} & \cdots & h_{n,2} & h_{n,1} \\
h_{n-1,n} & h_{n-1,n-1} & \cdots & h_{n-1,2} & h_{n-1,1} \\
0 & h_{n-2,n} & \cdots & h_{n-2,2} & h_{n-2,1} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & h_1 & h_{11}
\end{bmatrix}
\]

(11a)

and

\[
H^c_k = H^c_k = \begin{bmatrix}
e_{n,n} & e_{n,n-1} & \cdots & e_{n,1} \\
e_{n-1,n} & e_{n-1,n-1} & \cdots & e_{n-1,1} \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & e_{11}
\end{bmatrix}
\]

(11b)

with

\[
g(s) = g_{n+1}(s) (\alpha_{i,j}s^{i-1} + \alpha_{i,j}s^{i-2} + \alpha_{i,j}s^{i-3} + \ldots
\]

+ \alpha_{i,j} \xi + \alpha_{i,j} \theta), \quad i = 1, 2, \ldots, n + 1,
\]

(14)

where the coefficients \(\alpha_{i,j}, \beta_{i,j}, j = 1, 2, \ldots, i, i = 1, 2, \ldots, n + 1\), are given iteratively by

\[
\alpha_{i+1,j} = \beta_{i,j} + \sum_{k=1}^{j-1} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k})
\]

\[
\beta_{i,j} = \beta_{i,j} + \sum_{k=1}^{j} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad i = 2, 3, \ldots, n
\]

(15a)

and

\[
\beta_{i+1,j} = \beta_{i+1,j} + \sum_{k=1}^{j-1} (\beta_{i+1,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i+1,j,k+1,k})
\]

(15b)

with initial values

\[
\beta_{i,1} = 1, \quad \beta_{i,j} = e_{i,j}, \quad \beta_{i,j+1} = h_{i,j},
\]

(15c)

The proof of this theorem is provided in the Appendix.

In the case of \(H^c_k = I_n\), it is easy to show that the above

\[
\alpha_{i+1,j} = \alpha_{i,j} - \sum_{k=1}^{j-1} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad \alpha_{i+1,j} = 0,
\]

\[
\beta_{i+1,j} = \beta_{i+1,j} + \sum_{k=1}^{j} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad i = 2, 3, \ldots, n
\]

(16a)

with initial values

\[
\beta_{i,2} = \beta_{i,j+1} = h_{i,j}, \quad \beta_{i,1} = 1, \quad i = 1, 2, \ldots, n + 1.
\]

(16b)

This is the main result in [30]. (see also [29] and [31]).

### 3.2. The numerically stable iteration formula

Theorem 3.1 gives a very simple iterative formula

\[
d_{x}(s) = g_{n+1}(s)
\]

\[
N_{k}(s) = [g_{n}(s) \ g_{n-1}(s) \ \cdots \ g_{1}(s)]^{T}
\]

(13)

Theorem 3.1 gives a very simple iterative formula for the coefficients of a coprime factorization for a regular

\[
de_{x}(s) = h_{n+1}(s) (\alpha_{i,j}s^{i-1} + \alpha_{i,j}s^{i-2} + \alpha_{i,j}s^{i-3} + \ldots
\]

+ \alpha_{i,j} \xi + \alpha_{i,j} \theta), \quad i = 1, 2, \ldots, n + 1,
\]

(14)

where the coefficients \(\alpha_{i,j}, \beta_{i,j}, j = 1, 2, \ldots, i, i = 1, 2, \ldots, n + 1\), are given iteratively by

\[
\alpha_{i+1,j} = \beta_{i,j} + \sum_{k=1}^{j-1} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k})
\]

\[
\beta_{i,j} = \beta_{i,j} + \sum_{k=1}^{j} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad i = 2, 3, \ldots, n
\]

(15a)

and

\[
\beta_{i+1,j} = \beta_{i+1,j} + \sum_{k=1}^{j-1} (\beta_{i+1,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i+1,j,k+1,k})
\]

(15b)

with initial values

\[
\beta_{i,1} = 1, \quad \beta_{i,j} = e_{i,j}, \quad \beta_{i,j+1} = h_{i,j},
\]

(15c)

The proof of this theorem is provided in the Appendix.

In the case of \(H^c_k = I_n\), it is easy to show that the above

\[
\alpha_{i+1,j} = \alpha_{i,j} - \sum_{k=1}^{j-1} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad \alpha_{i+1,j} = 0,
\]

\[
\beta_{i+1,j} = \beta_{i+1,j} + \sum_{k=1}^{j} (\beta_{i,j,k} \alpha_{i+1,k+1,k} - \beta_{i+1,k} \alpha_{i,j,k+1,k}), \quad i = 2, 3, \ldots, n
\]

(16a)

with initial values

\[
\beta_{i,2} = \beta_{i,j+1} = h_{i,j}, \quad \beta_{i,1} = 1, \quad i = 1, 2, \ldots, n + 1.
\]

(16b)

This is the main result in [30]. (see also [29] and [31]).

### 3.2. The numerically stable iteration formula

This is the main result in [30]. (see also [29] and [31]).
and

\[
x^- = \begin{cases} 
-x, & \text{if } x < 0 \\
0, & \text{if } x \geq 0
\end{cases}.
\]

With these notations, it is clear that an arbitrary real scalar \( x \) can be expressed as \( x = x^+ - x^- \). Specifically, for the groups of numbers \( \tilde{e}_{i,j} \)'s and \( \tilde{h}_{i,j} \)'s defined in (12), we have

\[
\tilde{e}_{i,j} = \tilde{e}_{i,j}^+ - \tilde{e}_{i,j}^-, \quad \tilde{h}_{i,j} = \tilde{h}_{i,j}^+ - \tilde{h}_{i,j}^-,
\]

\( j = 1, 2, \ldots, i + 1, i = 1, 2, \ldots, n. \) (19)

The basic idea underlying the modification of the iterative formula (15) is to gather all the non-negative terms on the right-hand side of (15) together into one group and, gather all the negative terms together into another group, and then carry out the additions for these two groups of terms separately so as to avoid subtraction manipulations. Such a process involves two groups of non-negative scalars \( \alpha_{i,j}^+ \)'s and \( \alpha_{i,j}^- \)'s, which satisfy

\[
\alpha_{i,j} = \alpha_{i,j}^+ - \alpha_{i,j}^-, \quad j = 1, 2, \ldots, i + 1, \quad i = 1, 2, \ldots, n.
\]

(20)

Using (19) and (20) yields

\[
\tilde{e}_{i,j} \alpha_{i,j} = (\tilde{e}_{i,j}^+ - \tilde{e}_{i,j}^-) (\alpha_{i,j}^+ - \alpha_{i,j}^-) = (\tilde{e}_{i,j}^+ \alpha_{i,j}^+ + \tilde{e}_{i,j}^- \alpha_{i,j}^-) - (\tilde{e}_{i,j}^+ \alpha_{i,j}^- + \tilde{e}_{i,j}^- \alpha_{i,j}^+),
\]

\[
\sum_{k=1}^{j-1} \tilde{e}_{i,j-k,k} \alpha_{i,j-k,k} = \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ - \tilde{e}_{i,j-k}^- \alpha_{i,j-k}^-)
\]

\[
= \sum_{k=1}^{j-1} \left( \tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} + \tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k} \right)
\]

\[
= \sum_{k=1}^{j-1} \left( \tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} \right)
\]

\[
- \sum_{k=1}^{j-1} \left( \tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k} \right)
\]

(21)

and

\[
- \sum_{k=1}^{j-1} \tilde{h}_{i,j-k,k+1} \alpha_{i,j-k,k+1}
\]

\[
= \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} \right)
\]

\[
- \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1} \right)
\]

\[
- \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} \right)
\]

\[
= \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1} \right)
\]

\[
- \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} \right)
\]

(23)

Substituting (20)-(23) into (15), produces

\[
\alpha_{i+1,j}^+ - \alpha_{i+1,j}^- = (\tilde{e}_{i,j}^+ \alpha_{i,j}^+ + \tilde{e}_{i,j}^- \alpha_{i,j}^-)
\]

\[
+ \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} + \tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k})
\]

\[
+ \sum_{k=1}^{j-1} (\tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1})
\]

\[
- (\tilde{e}_{i,j}^+ \alpha_{i,j}^+ + \tilde{e}_{i,j}^- \alpha_{i,j}^-)
\]

\[
+ \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} + \tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k})
\]

\[
- \sum_{k=1}^{j-1} \left( \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1} \right)
\]

\( j = 2, 3, \ldots, i + 1, \quad i = 2, \ldots, n. \) (24)

This suggests that the scalars \( \alpha_{i,j}^+ \)'s and \( \alpha_{i,j}^- \)'s can be obtained through the following joint iteration procedure

\[
\begin{aligned}
\alpha_{i+1,j}^+ &= \tilde{e}_{i,j}^+ \alpha_{i,j}^+ + \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} + \tilde{h}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k+1}) \\
&\quad + \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1})
\end{aligned}
\]

\[
\alpha_{i+1,j}^- = \tilde{e}_{i,j}^- \alpha_{i,j}^- + \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^+ \alpha_{i,j-k}^+ \alpha_{i-j-k,k} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1}) \\
&\quad + \sum_{k=1}^{j-1} (\tilde{e}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k} + \tilde{h}_{i,j-k}^- \alpha_{i,j-k}^- \alpha_{i-j-k,k+1})
\]

\( j = 2, 3, \ldots, i, \quad i = 2, \ldots, n \) (25a)

and

\[
\begin{aligned}
\alpha_{i+1,i+1}^+ &= \sum_{k=1}^{i} (\tilde{h}_{i,i-k}^+ \alpha_{i,i-k}^+ \alpha_{i-k,i+k}) \\
\alpha_{i+1,i+1}^- &= \sum_{k=1}^{i} (\tilde{h}_{i,i-k}^- \alpha_{i,i-k}^- \alpha_{i-k,i+k}), \quad i = 2, 3, \ldots, n
\end{aligned}
\]

(25b)

In view of (15c), the initial values for the iteration procedure (25) are

\[
\begin{aligned}
\alpha_{1,1}^+ &= 1, \quad \alpha_{1,1}^- = 0 \\
\alpha_{2,1}^+ &= (e_{11})^+, \quad \alpha_{2,1}^- = (e_{11})^- \\
\alpha_{2,2}^+ &= (-h_{11})^+, \quad \alpha_{2,2}^- = (-h_{11})^-
\end{aligned}
\]

(25c)

Similar to the iterative formula (15), the joint itera-
tive procedure in (25) involves only simple scalar multiplication and addition manipulations and does not contain divisions. But different from (15), it does not contain subtraction manipulations either since all terms involved in formula (25) are non-negative when the initial values are taken as in (25c). Further, in view of the Remark 3.2 below, possible overflows can be easily prevented by resetting the values of $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$. Therefore, this iteration procedure is not only very simple but also numerically stable.

In the case of $H_k = I_n$, the above iteration (25) becomes

$$
\begin{aligned}
\alpha^e_{i+1,j} &= \alpha^e_{i,j} + \sum_{k=1}^{j-1} (\delta^e_{i-1,j-k+1} \alpha^e_{i-1,j-k+1} + \delta^x_{i-1,j-k+1} \alpha^x_{i-1,j-k+1}), \\
\text{with } \alpha^e_{1,1} &= 0,
\end{aligned}
$$

$$
\begin{aligned}
\alpha^x_{i+1,j} &= \alpha^x_{i,j} + \sum_{k=1}^{j-1} (\delta^e_{i-1,j-k+1} \alpha^x_{i-1,j-k+1} + \delta^x_{i-1,j-k+1} \alpha^x_{i-1,j-k+1}), \\
\text{with } \alpha^x_{1,1} &= 0
\end{aligned}
$$

$$
j = 2, 3, \ldots, i + 1, \quad i = 2, \ldots, n 
$$

(26a)

with the iteration initial values

$$
\begin{aligned}
\alpha^e_{1,2} &= (-h_{11})^+, \\
\alpha^x_{1,2} &= (-h_{11})^+
\end{aligned}
$$

$$
\begin{aligned}
\alpha^e_{i,1} &= 1, \quad \alpha^x_{i,1} = 0, \quad i = 1, 2, \ldots, n + 1
\end{aligned}
$$

(26b)

The following two remarks further states two problems involved with the usage of the iteration formula (25):

**Remark 3.1.** Note that the coefficients to the solution of the right coprime factorization are really given by (20). Although $\alpha^e_{i,j}$’s and $\alpha^x_{i,j}$’s are very accurate since they are given by the numerically stable procedure (25), if, for some $i$ and $j$, $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$ are very close, we will inevitably get a $a_{i,j}$ which may not be very accurate. In the case that the $a_{i,j}$’s are not the final interested results, it is recommended to carry on using $\alpha^e_{i,j}$’s and $\alpha^x_{i,j}$’s in the subsequent computation. This idea is demonstrated in the next subsection with the computation of $T_i N_i(s)$.

**Remark 3.2.** Since there are no subtractions involved, obviously, $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$ are getting larger and larger as the iteration goes on. For very large systems, this may cause an overflow in performing the iteration (25). Note that it is really the difference between $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$, that is, $\alpha^e_{i,j} = \alpha^e_{i,j} - \alpha^x_{i,j}$, that we are interested in rather than $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$ themselves, we can simply subtract a proper positive number $\kappa_i$ from both $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$ when they are getting very large. We call this process resetting the values of $\alpha^e_{i,j}$ and $\alpha^x_{i,j}$. In general, this number $\kappa_i$ can be chosen to be $\min(\alpha^e_{i,j}, \alpha^x_{i,j}) - \gamma$, with $0.1 \leq \gamma \leq 1$.

**3.3. Right coprime factorization of system ($EA_b$)**

Based on the relations in (7), Theorems 3.1 and the iteration (25), the following result about the right coprime factorization of a regular linear single-input descriptor system ($EA_b$) can be obtained (proof omitted).

**Theorem 3.2.** Let $(H_k, H_s, H_h)$ be the GHEF controllable subsystem of ($EA_b$) associated with the orthogonal matrix pair $(S, T)$. Then a solution to the right coprime factorization for the descriptor system ($EA_b$) is given by

$$
\begin{bmatrix}
N(s) \\
d(s)
\end{bmatrix} =
\begin{bmatrix}
z_1(s) \\
\vdots \\
z_s(s)
\end{bmatrix}
$$

(27)

with

$$
z_i(s) = \sum_{j=1}^{n+1} \beta_{i,j-1}s^{j-1}, \quad i = 1, 2, \ldots, s + 1
$$

(28)

where the coefficients $\beta_{i,j}$’s are given by

$$
\beta_{i,j} = \beta_{i,j}^e - \beta_{i,j}^x, \quad j = 0, 1, 2, \ldots, n, \quad i = 1, 2, \ldots, s + 1
$$

(29)

with

$$
\begin{aligned}
\beta_{i,j}^e &= \sum_{k=1}^{i-1} [t^e_{i,k-1,k+1} h_{i,n+1,k}^+ + t^e_{i,k-1,k+1} h_{i,n+1,k}^-] \alpha_{i,j}^e \\
+ [t^e_{i,k-1,k+1} h_{i,n+1,k}^+ + t^e_{i,k-1,k+1} h_{i,n+1,k}^-] \alpha_{i,j}^x \\
\beta_{i,j}^x &= \sum_{k=1}^{i-1} [t^x_{i,k-1,k+1} h_{i,n+1,k}^+ + t^x_{i,k-1,k+1} h_{i,n+1,k}^-] \alpha_{i,j}^e \\
+ [t^x_{i,k-1,k+1} h_{i,n+1,k}^+ + t^x_{i,k-1,k+1} h_{i,n+1,k}^-] \alpha_{i,j}^x \\
j &= 0, 1, 2, \ldots, n, \quad i = 1, 2, \ldots, s
\end{aligned}
$$

(30a)

and

$$
\begin{aligned}
\beta_{i,j}^e &= \alpha_{i+1,n+1}^e - \alpha_{i,j-1}^e, \quad \beta_{i,j}^x &= \alpha_{i+1,n+1}^x - \alpha_{i,j-1}^x, \\
j &= 1, 2, \ldots, n + 1
\end{aligned}
$$

(30b)

here $t_{ij}$ is the $i$-th row and the $j$-th column entry of the matrix $T$, and the $\alpha^e_{i,j}$’s and $\alpha^x_{i,j}$’s are iteratively given by (25).

It follows from Theorem 3.2 that a pair of solutions
Algorithm 3.1.

1. Solve a GUHF \((H^u_{g}, H^u_{s}, H^u_{0})\) of the given \(v\)-th order regular single-input linear descriptor system \((E \ A \ b)\) as well as the associated orthogonal matrices \(S\) and \(T\).
2. Find the GUHF controllable subsystem \((H_{g}, H_{s}, H_{0})\) of the system \((E \ A \ b)\), that is, the controllable subsystem of \((H^u_{g}, H^u_{s}, H^u_{0})\).
3. Solve \(\alpha_{ij}^g\) and \(\alpha_{ij}^s\), \(j = 1, 2, \ldots, i + 1, i = 1, 2, \ldots, n\) using iteration (25).
4. Solve a pair of solution \(N(s)\) and \(d(s)\) to the right coprime factorization of the descriptor system \((E \ A \ b)\) according to (27)-(30).

Remark 3.3. Obviously, the step 2 in the above algorithm does not involve numerical problems. Since the iteration (25) is numerically stable and the computation involved in (30) is also numerically stable, steps 3 and 4 are both numerically stable. If the step 1 of the above algorithm is performed by a numerically stable algorithm, for example, the ones in [1] and [22,24] (see also, [2]), the solution obtained through the above algorithm is then a numerically stable one.

Remark 3.4. The outputs of the above algorithm are either \(\beta_{ij}\) or \(\beta_{ij}^s\) and \(\beta_{ij}^N\), \(j = 0, 1, 2, \ldots, n, i = 1, 2, \ldots, v + 1\). If the pair of solutions \(N(s)\) and \(d(s)\) are the intermediate results for a certain problem, it is better to carry on the computation using \(\beta_{ij}^s\)'s and \(\beta_{ij}^N\)'s rather than \(\beta_{ij}\)'s obtained through (29). In the case that both \(\beta_{ij}^s\) and \(\beta_{ij}^N\) are very large, it may be necessary to reset their values (see, Remark 3.2).

IV. NUMERICAL EXAMPLES

Recalling from Theorem 3.1 that the solution to the right coprime factorization of the upper-triangular Hessenberg form \((H_{g}, H_{s}, H_{0})\) is given by (13)-(14). If we denote

\[
g_i(s) = g_{i1}s^{i-1} + g_{i2}s^{i-2} + g_{i3}s^{i-3} + \ldots + g_{i,i-1}s + g_{i,i},
\]

\(i = 1, 2, \ldots, n + 1\)  \hspace{1cm} (31)

then the solution can be expressed by the following matrix

\[
G = \begin{bmatrix}
g_{n+1,1} & g_{n+1,2} & \cdots & g_{n+1,n} & g_{n+1,n+1} \\
0 & g_{n,1} & \cdots & g_{n,n-1} & g_{n,n} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & g_{21} & g_{2,2} \\
0 & 0 & \cdots & 0 & g_{11}
\end{bmatrix}.
\]  \hspace{1cm} (32)

A Matlab program named factoriz has been worked out to realize the iteration (25). It produces the above \(G\) matrix when the system matrices \(H_{g}, H_{s}\) and \(H_{0}\) are input.

Example 1. Consider a system in the upper-triangular Hessenberg form with the following coefficient matrices:

\[
H_g = \begin{bmatrix}
6.4740 & -8.3450 & 79.4900 & 7.6050 & 0.9870 & 8.7700 \\
0 & 4.6570 & 5.5640 & 7.3960 & 7.4920 & 7.8900 \\
0 & 0 & -0.9980 & 4.5640 & 9.5400 & 9.3640 \\
0 & 0 & 0 & -7.4630 & -7.9770 & 3.7960 \\
0 & 0 & 0 & 0 & -9.8970 & 8.6970
\end{bmatrix},
\]

and

\[
H_s = \begin{bmatrix}
1.0000 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0000 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
\]

The above system \((H_{g}, H_{s})\) has been considered in [25] and [33] for pole assignment. Through applying our Matlab program factoriz, the solution to the right coprime factorization of the descriptor system is obtained as

\[
G = 1.0e+006 * 
\]

\[
\begin{array}{cccccccccccc}
0.09501 & 2.2311 & 0.4860 & 0.4565 & 10.6154 & 0.4057 \\
0 & 0.6068 & 0.8913 & 0.0185 & -0.7919 & 0.9355 \\
0 & 0 & -0.7621 & 0.8214 & 0.9218 & 0.9169 \\
0 & 0 & 0 & 0.4447 & 0.7382 & -5.4103 \\
0 & 0 & 0 & 0 & -0.1763 & 0.8936 \\
0 & 0 & 0 & 0 & 0 & 0.0000
\end{array}
\]

The solution to the right coprime factorization of the upper-triangular Hessenberg form \((H_{g}, H_{s}, H_{0})\) is given by (13)-(14). If we denote

\[
g_i(s) = g_{i1}s^{i-1} + g_{i2}s^{i-2} + g_{i3}s^{i-3} + \ldots + g_{i,i-1}s + g_{i,i},
\]

\(i = 1, 2, \ldots, n + 1\)  \hspace{1cm} (31)
In order to check the precision of the result, let us consider
\[
\epsilon = [(sH_E - H_d)N(s) - H_d d(s)]_{-1},
\]
\[
= (H_E - H_d)N(1) - H_d d(1), \quad (33)
\]
which may be viewed as an error vector, and an index can be taken as
\[
J = \epsilon^T \epsilon. \quad (34)
\]

For the above obtained result, we have \( J = 6.83259E-20 \).

**Example 2.** For the system considered in Example 1, let us only change the (6,6) element in \( H_E \) to 0.5. Applying our program `factoriz`, gives us a matrix \( G \) in the form of (32). Notice that in this case \( H_E \) is nonsingular, the (1,1) element of the obtained matrix \( G \) is different from zero. Dividing the matrix \( G \) by its (1,1) element, yields the following result which has a monic \( D(s) \):

\[
Gm = 1.0e + 008 \times
\begin{align*}
\text{Columns 1 through 4} \\
0.00000000010000 & 0.00000001272933 0.00001192626296 0.00033466133493 \\
0 0.00000001052521 & 0.0000118083120 0.00010879414986 \\
0 0 0.00000001229432 & 0.00001206199248 \\
0 0 0 0 -0.00000068620216 0.00000001539981 0.00000003968360 0.00000007630435 \\
0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0
\end{align*}
\]

The corresponding index \( J = 5.91695E-12 \). While with the result obtained using the `tf` command, \( J = 7.58636E-4 \).

**Example 4.** Consider a conventional system in upper-triangular Hessenberg form with the same coefficient matrices \( H_A \) and \( H_b \) as in Example 1, but with \( H_E \) being an identity matrix of order 6. Using `factoriz`, we obtain

\[
Gm = 1.0e + 006 \times
\begin{align*}
\text{Columns 1 through 4} \\
0.00000001000000 & 0.00000011955000 -0.00032989336100 0.000453879211475 \\
0 0.0000010000000 & 0.0000002503000 -0.00038125963000 \\
0 0 0.00000004740000 & 0.00000007225995200 \\
0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0
\end{align*}
\]

Since the matrix \( H_E \) is nonsingular, we can define
\[
A = H_E^{-1} H_A, \quad B = H_E^{-1} H_b.
\]
Further let \( C = I \) and \( D = 0 \), we can then obtain the right coprime factorization of the system by finding the transfer function of the system \((A, B, C, D)\). Using the Matlab command `tf` or `tfdata`, the result is obtained and found to be close to our result above, but is not as accurate since with our result the index given in (34) is \( J = 6.39394E-16 \), while with the result obtained using `tf` command the index \( J = 6.47514E-12 \).

**Example 3.** Using `tf` command to compute the right coprime factorization of a linear system in descriptor form with \( H_E \) nonsingular, the matrix \( H_E \) has to be inverted first. Obviously, when some diagonal element of \( H_E \) is small, it is likely that the `tf` command will produce poor result. To test this, let us again consider the system in Example 1, but with the (6,6) element in \( H_E \) replaced by 0.01. In this case, by applying our program `factoriz`, we obtain the following result:

\[
Gm = 1.0e + 010 \times
\begin{align*}
\text{Columns 1 through 4} \\
0 & 0.00000001012987 0.0000005748155561 0.00017095573306 \\
0 0.00000000105252 0.000000439553082 0.000005103041547 \\
0 0 0.00000001229432 0.000000486402719 \\
0 0 0 0 -0.000000006862020 \\
0 0 0 0 0 \\
0 0 0 0 0
\end{align*}
\]

With this solution, \( J = 4.12728E-19 \). While with the solution obtained using the `tf` command, \( J = 7.36595E-17 \). All the computation above was done with Matlab 5.3. It is seen from the above examples that the iteration formula proposed in this paper is indeed more numerically favorable.
V. CONCLUSION

This paper presents a simple, efficient numerically stable solution to the right coprime factorization of a regular single-input descriptor linear system. It is shown that when the given single-input system is converted into an upper-triangular Hessenberg form by some orthogonal transformations, the solution to the right coprime factorization of the system can be easily obtained by a scalar iterative formula which gives the coefficients of the solution to the right coprime factorization of the system. This formula uses directly the parameters of the Hessenberg form and involves only multiplication and additions of non-negative scalar terms. Numerical examples show the effect and numerical advantage of the presented formula.

ACKNOWLEDGEMENTS

This work was supported in part by the Outstanding Youth Foundation of the Chinese Ministry of Education and the Chinese National Natural Science Foundation under Grant No.69504002.

The author is grateful to the reviewers for their helpful comments and suggestions.

APPENDIX. PROOF OF THEOREM 3.1

A. The preliminary lemma

In order to prove Theorem 3.1, a simple method proposed and used in [13] for solving the right coprime factorizations of linear descriptor systems based on matrix elementary transformations is stated.

It is obvious that, when the descriptor linear system \((E\ A\ b)\) is controllable, there exist a pair of unimodular polynomial matrices \(P(s)\) and \(Q(s)\) of dimensions \(n\times n\) and \((n+1)\times(n+1)\), respectively, such that

\[
P(s)[b\ A - sE]Q(s) = [I\ 0].
\] (A1)

Based on this relation, a solution to right coprime factorization of descriptor system \((E\ A\ b)\) can be given following the method in [13].

Lemma A1. Let \((E\ A\ b)\) be controllable and regular, and \(Q(s)\) be the unimodular matrix of dimension \((n+1)\times(n+1)\) satisfying (A1). Then a solution to the right coprime factorization for the descriptor system \((E\ A\ b)\) is given by

\[
\begin{bmatrix}
  d(s) \\
  N(s)
\end{bmatrix} = \alpha Q(s)
\begin{bmatrix}
  0_n \\
  1
\end{bmatrix},
\] (A2)

where \(\alpha\) is an arbitrary nonzero scalar.

B. The proof of Theorem 3.1

The proof of Theorem 3.1 is composed of three steps. In the first step, we prove the following result based on Lemma A1.

Lemma B1. Let \((H_1\ H_2\ H_3)\), given by (11), be a single-input descriptor linear system in upper-triangular Hessenberg form. Then a solution to the right coprime factorization (8) for the system \((H_1\ H_2\ H_3)\) is given by

\[
\begin{bmatrix}
  d_h(s) \\
  N_h(s)
\end{bmatrix} = Q_n(s)Q_{n-1}(s)\cdots Q_1(s)
\begin{bmatrix}
  0_n \\
  1
\end{bmatrix}
\] (B1)

with

\[
Q(s) =
\begin{bmatrix}
  h_1 \\
  : \\
  : \\
  h_i \\
  : \\
  : \\
  h_i \\
  \vdots \\
  h_i \\
\end{bmatrix}_{(n+1)\times(n+1)}
\] (B2)

Proof. Note that

\[
[H_1\ H_2 - sE]
\]

\[
= \begin{bmatrix}
  h_1 & h_{n+1} - se_1 & \cdots & h_{n+1} - se_{n+1} & h_{n+1} - se_n \\
  0 & h_{n+1} & \cdots & h_{n+1} - se_{n+1} & h_{n+1} - se_n \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & \cdots & h_2 & h_1 - se_{n+1}
\end{bmatrix}_{(n+1)\times(n+1)}
\] (B3)

Due to the special structure of the above matrix and the property of \(h_i \neq 0, i = 1, 2, \ldots, n\), it is easy to show that a pair of unimodular matrices \(P(s)\) and \(Q(s)\) satisfying

\[
P(s)[H_1\ H_2 - sE]Q(s) = [I\ 0]
\]

are given by

\[
P(s) = \text{diag}[h_1^{-1} \ h_2^{-1} \ \cdots \ h_n^{-1}]
\] (B4)

and
\[ Q(s) = \frac{1}{h_{\lambda_2} \cdots h_n} Q_n(s) Q_{n-1}(s) \cdots Q_1(s), \quad (B5) \]

where \( Q_i(s), i = 1, 2, \ldots, n, \) are as defined in (B2).

Using Lemma A1 and choosing \( \alpha = h_{\lambda_2} \cdots h_n, \) gives the formula (B1).

In the second step, we prove the following lemma which converts further the result in Lemma B1 into a simple polynomial iterative formula.

**Lemma B2.** Let \( (H_E, H_A, H_0) \), given by (11), be a single-input descriptor linear system in upper-triangular Hessenberg form. Then a solution to the right coprime factorization (8) for the system \( (H_E, H_A, H_0) \) is given by

\[
\begin{align*}
&d_n(s) = f_n(s), \\
&N_n(s) = [f_{n+1}(s) ~ f_n(s) ~ f_{n-1}(s) ~ \cdots ~ f_1(s)]^T
\end{align*}
\]

with the polynomials \( f_i(s), i = 1, 2, \ldots, n+1, \) given iteratively by

\[
\begin{align*}
&f_1(s) = 1, \\
&f_2(s) = s - h_{11}, \\
&f_{i+1}(s) = \sum_{j=1}^{i} h_{i,j} f_j(s), \quad i = 2, 3, \ldots, n.
\end{align*}
\]

**Proof.** Due to the structure of \( Q_i(s), \) and by using (B7), we first have

\[
Q_1(s) = \begin{bmatrix} 0_{n-1} \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{n-1} \\ s e_{11} - h_{11} \end{bmatrix} = \begin{bmatrix} 0_{n-1} \\ f_2(s) \\ h_1 f_1(s) \end{bmatrix}.
\]

Based on the above, and again using (B7), we further have

\[
Q_{n-1}(s) Q_n(s) \begin{bmatrix} 0_{n-1} \\ 1 \end{bmatrix} = Q_{n-1}(s) \begin{bmatrix} 0_{n-1} \\ f_2(s) \\ h_1 f_1(s) \end{bmatrix} = \begin{bmatrix} 0_{n-2} \\ e_{22}s - h_{22} \end{bmatrix} = \begin{bmatrix} 0_{n-2} \\ f_3(s) \\ h_2 f_2(s) \\ h_1 f_1(s) \end{bmatrix} + \begin{bmatrix} 0_{n-2} \\ e_{21}s - h_{21} \end{bmatrix} = \begin{bmatrix} 0_{n-2} \\ f_4(s) \\ h_3 f_3(s) \\ h_2 f_2(s) \\ h_1 f_1(s) \end{bmatrix}.
\]

Thus, the conclusion follows immediately from Lemma B1.

**Lemma B2** is a direct extension to the main result in [28]. Based on Lemma B2, we can now, in the final step, derive the set of iterative formulae (13)-(15) in Theorem 3.1.
Using the notations introduced in (12b), the iteration in (B7) can be written as

\[ f_{i+1}(s) = \sum_{j=1}^{n} (\bar{e}_{i,j} s - \bar{\alpha}_{i}) f_{j}(s) \]

\[ = \sum_{j=1}^{n} \bar{e}_{i,j} f_{j}(s) - \sum_{j=1}^{n} \bar{\alpha}_{i,j} f_{j}(s), \quad i = 2, 3, \ldots, n. \]  

(B8)

Let

\[ f_{i}(s) = \alpha_i s^{i-1} + \alpha_{i+1} s^{i-2} + \alpha_{i+2} s^{i-3} + \ldots \]

\[ + \alpha_{i+1} s + \alpha_{i+1, 1} i = 1, 2, \ldots, n + 1 \]  

(B9)

then, (13) and (14) can be readily obtained from (B6) and (B9).

Denote

\[ d_{i}(s) = \sum_{j=1}^{n} \bar{h}_{i,j} f_{j}(s) \]

\[ = d_{i, 1} s^{i-1} + d_{i, 2} s^{i-2} + \ldots + d_{i, i-1} s + d_{i, i} \]  

(B10)

and

\[ r_{i}(s) = \sum_{j=1}^{i} \bar{e}_{i,j} f_{j}(s) \]

\[ = r_{i, 1} s^{i-1} + r_{i, 2} s^{i-2} + \ldots + r_{i, i-1} s + r_{i, i} \]  

(B11)

then (B8) becomes

\[ f_{i+1}(s) = s r_{i}(s) - d_{i}(s), \quad i = 2, 3, \ldots, n. \]  

(B12)

Noticing that

\[ f_{i+1}(s) = \alpha_{i+1} s^{i} + \alpha_{i+1, 1} s^{i-1} + \alpha_{i+1, 2} s^{i-2} \]

\[ + \ldots + \alpha_{i+1} s + \alpha_{i+1, 1} s^{i-1} \]

Comparing the coefficients of \( s' \) on both sides of the equations in (B12) gives

\[ \alpha_{i+1, 1}(s) = r_{i, 1} - d_{i, 1}, \quad j = 1, 2, \ldots, i; \quad i = 1, 2, \ldots, n \]  

(B13a)

and

\[ \alpha_{i+1, i+1}(s) = -d_{i, i}, \quad i = 1, 2, \ldots, n \]  

(B13b)

In view of (B10) and

\[ \bar{h}_{i,j} f_{j}(s) = \bar{h}_{i,j} \alpha_{j} s^{j-1} + \bar{h}_{i,j} \alpha_{j+1} s^{j-2} + \bar{h}_{i,j} \alpha_{j+2} s^{j-3} \]

\[ + \ldots + \bar{h}_{i,j} \alpha_{j+1} s + \bar{h}_{i,j} \alpha_{j} s^{j-1} \]

the coefficients of polynomial \( d(s) \) can be obtained as

\[
\begin{align*}
    d_{i, 1} &= \bar{h}_{i, i} \alpha_{i, 1} \\
    d_{i, 2} &= \bar{h}_{i, i} \alpha_{i, 2} + \bar{h}_{i, i-1} \alpha_{i, 1} \\
    & \vdots & \\
    d_{i, i-1} &= \bar{h}_{i, i} \alpha_{i, i-1} + \bar{h}_{i, i-1} \alpha_{i, i-2} + \ldots + \bar{h}_{i, 1} \alpha_{i, 1} \\
    d_{i, i} &= \bar{h}_{i, i} \alpha_{i, i} + \bar{h}_{i, i-1} \alpha_{i, i-1} + \bar{h}_{i, i-2} \alpha_{i, i-2} + \ldots + \bar{h}_{i, 1} \alpha_{i, 1}
\end{align*}
\]

which can be compactly written as

\[ d_{i,j} = \sum_{k=1}^{j} \bar{h}_{i,j-k+1,k} \alpha_{i-j+k,k}, \quad j = 1, 2, \ldots, i, \]

\[ i = 2, 3, \ldots, n. \]  

(B14)

Similarly, the coefficients of polynomial \( r(s) \) are given as

\[ r_{i,j} = \sum_{k=1}^{i} \bar{e}_{i,j-k+1,k} \alpha_{i-j+k,k}, \quad j = 1, 2, \ldots, i, \]

\[ i = 2, 3, \ldots, n. \]  

(B15)

Substituting (B14) and (B15) into (B13a), yields

\[ \alpha_{i+1, i}(s) = \sum_{k=1}^{i} \bar{e}_{i,j-k+1,k} \alpha_{i-j+k,k} \]

\[ - \sum_{k=1}^{i} \bar{h}_{i,j-k+1,k} \alpha_{i-j+k,k}, \quad i = 1, 2, \ldots, n, \]

(B16)

which is equivalent to the iterative formula (15a). Substituting (B14) into (B13b), yields (15b). Finally noticing that the initial values are obviously given as in (15c), the whole proof of Theorem 3.1 is then complete.

REFERENCES


Guang-Ren Duan was born in Heilongjiang Province, P. R. China, on April 5, 1962. He received his BSc. degree in Applied Mathematics, and both his MSc and PhD degrees in Control Systems Theory. From 1989 to 1991, he was a post-doctoral researcher at Harbin Institute of Technology, China, where he became a professor of control systems theory in November 1991. Since December 2000, he has been elected Specially Employed Professor at Harbin Institute of Technology sponsored by the Cheung Kong Scholars Program of the Chinese government. He is now the Director of the Center for Control Theory and Guidance Technology at Harbin Institute of Technology. Dr Duan visited the University of Hull, UK, and the University of Sheffield, UK from December 1996 to October 1998. He is currently on leave at the Queen’s University of Belfast, UK. Dr Duan is the author of over 220 publications. His main research interests include robust control, eigenstructure assignment, descriptor systems, missile autopilot control and magnetic bearing control. Dr Duan is a Chartered Engineer in the UK, a Senior Member of IEEE and a Fellow of IEE.