87-08

INT. J. CONTROL, 1987, VOL. 45, NO. 4, 1137–1146

Realization algorithm for determining generalized state-space representations

SHAOHUA TAN† and JOOS VANDEWALLE†

A realization algorithm is presented that determines a generalized state-space representation from a so-called column-pseudoproper right MFD. This algorithm is a generalization of the realization algorithm given by Wolovich and is proved to be controllable in the Rosenbrock sense. Moreover, the dimension of the realization is equal to the determinantal degree of the given right MFD in a generalized sense.

1. Introduction

The main purpose of this paper is to present a direct and constructive realization algorithm that determines a controllable generalized state-space system of the form

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{1 a}$$

$$y(t) = Cx(t) \tag{1 b}$$

from a transfer matrix $G(s)$ in right matrix-fraction-description (MFD), i.e.

$$G(s) = N(s)D^{-1}(s) \tag{2}$$

In the above, $x(t)$ is an n-vector of internal variables, $u(t)$ is an m-vector of control inputs and $y(t)$ is a p-vector of outputs. E, A, B, C are real matrices of appropriate dimensions. E may be singular. $G(s)$ is a $p \times m$ rational matrix; $N(s)$ and $D(s)$ are polynomial matrices of dimensions $p \times m$ and $m \times m$ respectively. Furthermore, $D(s)$ is non-singular for almost all s; $N(s)$ and $D(s)$ may have common right-divisors.

The algorithm that will be developed here is an interesting generalization of the realization algorithm presented by Wolovich (1973) (see also Kailath (1980), p. 403, for an exposition in relation to computer simulation). As shown by Wolovich (1973) and Kailath (1980), in order to realize the given right MFD into a state-space representation (i.e. $E = I$ in (1 a), where I is the identity matrix), two conditions have to be satisfied by (2), namely, $G(s)$ must be strictly proper and the polynomial matrix $D(s)$ must be column-reduced. However, since (2) may result from physical modelling or from some kinds of equivalent transformations, these two conditions, especially the column-reducedness, are seldom met. Of course, by applying the division process the strict proper part can be separated out, and by employing unimodular matrices the column-reducedness can be achieved. In doing so, however, the structural properties of (2) at infinity are completely altered. There are two reasons for this. First, as pointed out by Rosenbrock (1974) (see also Rosenbrock and Pugh 1974), the non-proper part also contains the structural properties analogous to the proper part, which cannot be accommodated by the state-space representations. Secondly, unimodular matrices can destroy the zero structures of both $N(s)$ and $D(s)$ at infinity.

In contrast with Wolovich's algorithm, this new realization algorithm puts more

Received 9 June 1986.

† ESAT Laboratory, Department of Electrical Engineering, K. U. Leuven, Kardinaal Mercierlaan 94, B-3030 Heverlee, Belgium.

emphasis on preserving the structure of the given right MFD at infinity. Although similar constructions for matrices A, B, C are used, an additional matrix E is generated, which takes into consideration the structure of (2) at infinity. Thereby, the column-reducedness assumption is dropped, and the strict properness is replaced by a weaker condition, namely column-pseudoproperness. Moreover, the realized system is proven to be controllable, and, with the appropriate definition of the determinantal degree of polynomial matrices, it can be shown that $\det D(s) = \det(Es - A)$. These well-known properties of Wolovich's algorithm all reappear with respect to a set of new definitions, whereas certain limitations are naturally removed.

Paramount to the development of this algorithm are recently obtained results concerning polynomial matrices, especially the properties related to the zero structure at infinity. Therefore some of these results that are necessary for our development are stated in § 2, while the presentation of the algorithm is deferred to § 3. In § 4 the verification and proofs of the algorithm and its properties are given. An example is used there to demonstrate the algorithm. In the last section this algorithm is compared to two recently developed algorithms in Bosgra and Van der Weiden (1981) and Christodoulou and Mertzios (1985), both for obtaining the generalized state-space representations. The applications of our algorithm are also indicated in the concluding remarks.

2. Preliminaries

This section contains some results concerning polynomials and polynomial matrices that are needed in the development of the realization algorithm.

Let $F_n[s]$ denote the set of all polynomials of the form

$$p(s) = p_0 s^n + p_1 s^{n-1} + \ldots + p_n \tag{3}$$

where each coefficient p_i $(i = 0, \ldots, n)$ belongs to a field F (in this paper F is taken to be the real field). It should be emphasized that $p_0 \neq 0$ is not required, hence essentially all polynomials of the form (3) are in $F_n[s]$. n will be called the *generalized degree* of the polynomial (3); this term is coined to distinguish it from the normal definition of polynomial degree.

Within the set $F_n[s]$, polynomial equivalence, addition and scalar multiplication are defined as follows.

Let $a(s) = \sum_{i=0}^{n} a_i s^{n-i}$ and $b(s) = \sum_{i=0}^{n} b_i s^{n-i}$ be two elements of $F_n[s]$ and let c $(\neq 0)$ be a scalar from the field F; then we have the following definitions:

$$\text{equivalence:} \quad a(s) = b(s) \quad \text{if } a_i = b_i \quad (i = 0, \ldots, n)$$

$$\text{addition:} \quad a(s) + b(s) = \sum_{i=0}^{n} (a_i + b_i) s^{n-i}$$

$$\text{scalar multiplication:} \quad c \cdot a(s) = \sum_{i=0}^{n} c \cdot a_i s^{n-i}$$

Moreover, if a polynomial $g(s) = \sum_{i=0}^{k} g_i s^{k-i}$ in $F_k[s]$ is given then the product of $a(s)$ and $g(s)$ is defined by

$$\text{polynomial multiplication:} \quad a(s) \cdot g(s) = \sum_{i=0}^{n+k} r_i s^{n+k-i}$$

where $r_i = \sum_{t+j=i} a_t g_j \quad (i = 0, \ldots, n+k)$.

Note that $a(s) \cdot g(s)$ is an element of $F_{n+k}[s]$. With the above definitions of operations, it is a simple exercise to check that $F_n[s]$ is a linear space; furthermore, the commutative, associative and distributive laws of various kinds hold.

To avoid possible confusion, two remarks are in order. First, for two polynomials of different sets (say, one from $F_n[s]$, another from $F_k[s]$, $n \neq k$) the equivalence and addition are not defined. For example, if $p_1(s) = 0 \cdot s^2 + s + 1 \in F_2[s]$ and $p_2(s) = s + 1 \in F_1[s]$ be polynomials from two different sets then expressions like $p_1(s) = p_2(s)$ and $p_1(s) + p_2(s)$ are meaningless. Secondly, a non-zero scalar $d \in F$ should not be confused with the polynomial $p(s) = 0 \cdot s^n + 0 \cdot s^{n-1} + \ldots + 0 \cdot s + d$, which is an element of $F_n[s]$.

As consequences of the preceding seemingly trivial modifications, many interesting properties result, which lead to natural generalizations of some of the well-known classical results (e.g. coprimeness and the Sylvester resultant) so that a deeper insight into polynomials is gained. For further details see Tan and Vandewalle (1986). Here we shall quote one result that will be used later: any polynomial $p(s)$ in $F_n[s]$ has precisely n zeros, both finite and at infinity. The finite zeros are defined as usual. The zeros at infinity and their multiplicities are defined to be the zeros at the origin of the s-plane of the polynomial $\bar{p}(s) = s^n p(1/s)$, and their multiplicities.

We now turn to polynomial matrices. A polynomial matrix can be constructed in many different ways from the polynomial sets just defined. For instance, $p \times m$ polynomials can all be taken from $F_n[s]$, then arranged into a $p \times m$ matrix. A proper rearrangement shows that it can be rewritten as

$$P_0 s^n + P_1 s^{n-1} + \ldots + P_n \tag{4}$$

where each P_i $(i = 0, \ldots, n)$ is a $p \times m$ constant matrix. Although only implicit, this construction is widely used (see, e.g. Gantmacher (1960, p. 130)).

For convenience, a polynomial matrix constructed in this way is referred to as a matrix polynomial. In this paper, a slightly different construction is used. To be precise, let a $p \times m$ polynomial matrix be constructed. First choose p polynomials from $F_{k_1}[s]$ and arrange them into a column

$$\begin{bmatrix} a_1(s) \\ a_2(s) \\ \vdots \\ a_p(s) \end{bmatrix} \tag{5}$$

It is easily seen that (5) can be written as follows

$$\begin{bmatrix} a_{01} \\ a_{02} \\ \vdots \\ a_{0p} \end{bmatrix} s^{k_1} + \begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1p} \end{bmatrix} s^{k_1 - 1} + \ldots + \begin{bmatrix} a_{k_1 1} \\ a_{k_1 2} \\ \vdots \\ a_{k_1 p} \end{bmatrix} \tag{6}$$

where each column in (6) is a constant p-vector. We then proceed to choose p polynomials from $F_{k_2}[s]$, arrange them into a column like (5), and so on until m columns are constructed. Finally, the $p \times m$ polynomial matrix is formed by arranging these m columns in a row. In the above, $k_1, \ldots, k_m$, which will be called *column degrees*, may all be different. A polynomial matrix constructed in this way will be called in the column-based form. It is obvious that a polynomial matrix in the column-based form

is not necessarily a matrix polynomial (since it cannot be written in the form (4) unless $k_1, \ldots, k_m$ are all the same). Hence the matrix polynomial is a special case of the column-based form.

Equivalently, the question of whether a given polynomial matrix is either a matrix polynomial or a polynomial matrix in the column-based form, or neither, depends purely on the fact that its entries fit into the characterization of these forms. In practice, however, a common tendency is to skip leading terms with zero coefficients when writing down a polynomial. For example, the polynomial matrix

$$D(s) = \begin{bmatrix} 0 \cdot s^3 + 0 \cdot s^2 + s + 0 & s^2 + 0 \cdot s + 0 \\ 0 \cdot s^3 + 0 \cdot s^2 + 0 \cdot s + 1 & 0 \cdot s^2 + 0 \cdot s + 1 \end{bmatrix} \tag{7}$$

which is in the column-based form, is always written in short as

$$\tilde{D}(s) = \begin{bmatrix} s & s^2 \\ 1 & 1 \end{bmatrix} \tag{8}$$

This kind of practice places obstacles in deciding in which form the given polynomial matrix might be. For simplicity, we shall assume that all polynomial matrices encountered in this paper are in the column-based form. It might well be asked whether there is any special significance in the above lengthy definitions and descriptions. Indeed there is. Roughly speaking, these definitions ensure algebraically that the operations involved preserve not only the finite zeros but also the zeros at infinity. Thereby a unification between finite zeros and zeros at infinity is achieved. Since this is not the main concern of this paper, interested readers are again referred to Tan and Vandewalle (1986) for further details. As a simple example, when the determinant of the matrix $D(s)$ in (7) is to be computed, the usual method uses (8) directly, and thus gives the result as

$$\det D(s) = -s^2 + s \tag{9}$$

which is a polynomial of $F_2[s]$. Note that no zeros at infinity appear in (9). However, if we start the calculation from (7) and observe all rules of the operations defined above, we obtain

$$\det D(s) = 0 \cdot s^5 + 0 \cdot s^4 + 0 \cdot s^3 - s^2 + s + 0 \tag{10}$$

which is a polynomial of $F_5[s]$. Obviously, in (10) there are three zeros at infinity which are missing in (9). In general, it is easily verified that the generalized determinantal degree of a polynomial matrix in the column-based form is equal to the sum of its column degrees.

It may seem tedious to perform the calculation shown in the preceding example. In fact, it can be done in a much more efficient way by simply observing that terms with non-zero coefficients are not affected by leading zero terms; thus they can be computed conventionally (as in (9)). Then leading zero-coefficient terms can be added to the desired degree (which is usually obvious). This property is called the *separation principle.*

As a final point, a definition conerning MFD is given, which is needed in the next section. Let a right MFD be given as in (2), and assume that $N(s)$ and $D(s)$ are both in the column-based form. Denoting by $k_1, \ldots, k_m$ the column degrees of $D(s)$, and by $r_1, \ldots, r_m$ the column degrees of $N(s)$; then (2) is called column-pseudoproper if $r_i < k_i$ $(i = 1, \ldots, m)$.

3. Realization algorithm

With the preceding preliminary results, which are essential for the subsequent development, we are able to give a full description of the realization algorithm.

Consider a given column-pseudoproper right MFD (2), the aim is to find matrices E, A, B, C as in (1) such that $N(s)D^{-1}(s) = C(Es - A)^{-1}B$. Note that no conditions like column-reducedness or properness (in the normal sense) are required.

First decompose $D(s)$ in the form

$$D(s) = D_{hc}S(s) + D_{lc}T(s) \tag{11}$$

where D_{hc} denotes the highest-column-degree coefficient matrix, D_{lc} denotes the lowest-column-degree coefficient matrix and $S(s)$, $T(s)$ are defined as

$$S^{T}(s) = \begin{bmatrix} s^{k_1}s^{k_1-1}\dots s & 0 & \dots & 0 \\ 0 & s^{k_2}s^{k_2-1}\dots s & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & s^{k_m}s^{k_m-1}\dots s \end{bmatrix}$$

$$= \text{block diag}\,\{[s^{k_1}\dots s] \quad [s^{k_2}\dots s] \quad \dots \quad [s^{k_m}\dots s]\}$$

$$T^{T}(s) = \begin{bmatrix} s^{k_1-1}\dots s\ 1 & 0 & \dots & 0 \\ 0 & s^{k_2-1}\dots s\ 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ddots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & s^{k_m-1}\cdots s\ 1 \end{bmatrix}$$

$$= \text{block diag}\,\{[s^{k_1-1}\dots 1] \quad [s^{k_2-1}\dots 1] \quad \dots \quad [s^{k_m-1}\dots 1]\} \tag{12}$$

In the above, the superscript T denotes transpose. By construction it follows that $S(s) = T(s)\cdot s$. It should be noted that D_{hc} and D_{lc} are not unique in (11).

By the assumption of column-pseudoproperness and the structure of $T(s)$, the numerator $N(s)$ in (2) can be decomposed in the form

$$N(s) = N_c T(s) \tag{13}$$

We first construct a so-called *core realization*, which is based on the set of column degrees of $D(s)$, i.e. $k_1, k_2, \dots, k_m$. This can be done as follows:

$$E_c = I_n, \quad C_c = I_n$$

$$A_c = \text{block diag}\left\{\underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}}_{k_i}\right\} k_i,\ i = 1, \dots, m$$

$$B_c^T = \text{block diag}\{[\underbrace{1 \quad 0 \quad \ldots \quad 0}_{k_i}], i = 1, \ldots, m\} \tag{14}$$

where $n = \sum_{i=1}^{m} k_i$ is the generalized determinantal degree of $D(s)$. With the structures in (14), a realization of (2) in the form of (1) can be obtained as follows:

$$\left.\begin{aligned} E &= E_c - B_c B_c^T + B_c D_{hc} = I_n - B_c B_c^T + B_c D_{hc} \\ A &= A_c - B_c D_{lc} \\ B &= B_c \\ C &= N_c C_c = N_c \end{aligned}\right\} \tag{15}$$

It may be noted that the constructions of the matrices A, B, C are exactly the same as the constructions given, for example, in Kailath (1980). Moreover, the highest-column-degree coefficient matrix D_{hc} enters naturally, as a differential feedback gain, to form the matrix E.

We shall show in the next section that not only has the generalization been achieved by (15) but also the most important properties of Wolovich's algorithm are inherited.

4. Proof of the algorithm and its properties

We now show that (15) is indeed a realization of the given right MFD (2). By definition of the realization, if (15) realizes (2) the following must be true:

$$C(Es - A)^{-1} B = N(s) D^{-1}(s) \tag{16}$$

(Of course, we have to prove that $Es - A$ is non-singular. However, a proof for this will be delayed for a while.) Equivalently, by substituting (15) and (13) into (16), we have

$$N_c(Es - A)^{-1} B = N_c T(s) D^{-1}(s) \tag{17}$$

From (17), it is clear that if we can prove

$$(Es - A)^{-1} B = T(s) D^{-1}(s) \tag{18}$$

then (16) will follow immediately. In fact, this can be done quite easily as follows. First multiplying both sides of (18) on the left by $Es - A$ and on the right by $D(s)$,

$$B \cdot D(s) = (Es - A) \cdot T(s) \tag{19}$$

By substituting (15) and (11) into (19),

$$B[D_{hc} S(s) + D_{lc} T(s)] = [(I - BB^T + BD_{hc})s - A_c + BD_{lc}]T(s) \tag{20}$$

Noting that $S(s) = T(s) \cdot s$, (20) can be simplified as

$$S(s) - BB^T S(s) = A_c \cdot T(s) \tag{21}$$

Now (21) can be readily checked from the structures of the matrices $S(s)$, $T(s)$, B, A_c. It follows that (18) is true; hence (15) indeed realizes the given right MFD (2).

Two important properties of Wolovich's algorithm are that the realization is controllable and there is a direct relationship between the determinant of the denominator and the characteristic polynomial of the realization. We now show that these two properties are preserved (in the generalized sense) by the realization just developed.

Recall, first, that a generalized state-space system (1) is controllable if rank $[E \quad B] = n$ and rank $[Es - A \quad B] = n$ (Rosenbrock 1974). We now show that the generalized state-space system defined by (15) is controllable. The proof is straightforward. From the structures of the matrices E_c, A_c, B_c, it is readily checked that

$$\left.\begin{aligned} \text{rank } [E_c \quad B_c] &= n \\ \text{rank } [E_c s - A_c \quad B_c] &= n \end{aligned}\right\} \tag{22}$$

As a consequence of (22),

$$\begin{aligned} \text{rank } [E \quad B] &= \text{rank } [E_c + B_c(D_{hc} - B_c^T) \quad B_c] \\ &= \text{rank } [E_c \quad B_c] \begin{bmatrix} I & 0 \\ D_{hc} - B_c^T & I \end{bmatrix} = \text{rank } [E_c \quad B_c] = n \end{aligned} \tag{23 a}$$

and for all finite s,

$$\begin{aligned} \text{rank } [Es - A \quad B] &= \text{rank } [(E_c - B_c B_c^T + B_c D_{hc})s - A_c + B_c D_{lc} \quad B_c] \\ &= \text{rank } [E_c s - A_c \quad B_c] \begin{bmatrix} I & 0 \\ (D_{hc} - B_c^T)s + D_{lc} & I \end{bmatrix} \\ &= \text{rank } [E_c s - A_c \quad B_c] = n \end{aligned} \tag{23 b}$$

where I denotes the identity matrix of appropriate dimension. This proves our assertion.

For the second property, we prove

$$\det D(s) = \det (Es - A) \tag{24}$$

Note that, since the matrix $D(s)$ is non-singular, (24) implies that the matrix $Es - A$ is also non-singular. This assertion is needed in the proof of the realization algorithm. It should be emphasized that from the results in § 2, (24) implies that two polynomials are in the same polynomial set (with equal generalized degree).

By applying the separation principle, we only have to show first that two polynomials in (24) have the equal generalized degree and then to show that (24) is true in the normal sense.

First, the generalized degree of the polynomial $\det (Es - A)$ is obviously equal to n $\left(= \sum_{i=1}^{m} k_i\right)$, since every polynomial in $Es - A$ is an element of $F_1[s]$. Therefore it is equal to the generalized degree of the polynomial $\det D(s)$, which is also n. We can then prove (24) without being concerned about degrees. This is done as follows.

Observe first that

$$\{T(s), D(s)\} \text{ are right-coprime} \tag{25 a}$$

$$\{Es - A, B\} \text{ are left-coprime} \tag{25 b}$$

The relation (25 a) follows from the fact that

$$\text{rank} \begin{bmatrix} T(s) \\ D(s) \end{bmatrix} = m$$

because of the structure of $T(s)$ (see (12)), while (25 b) is a direct consequence of (23 b). It then follows from (25 a, b) that there exist polynomial matrices $\{X(s), Y(s)\}$ and

$\{\bar{X}(s), \bar{Y}(s)\}$ such that

$$(Es - A)X(s) + BY(s) = I \tag{26 a}$$

$$\bar{X}(s)T(s) + \bar{Y}(s)D(s) = I \tag{26 b}$$

where the identities I in (26) have appropriate dimensions. Equations (26 a, b) can be combined together with (19) in matrix form:

$$\begin{bmatrix} Es - A & B \\ -\bar{X}(s) & \bar{Y}(s) \end{bmatrix} \begin{bmatrix} X(s) & -T(s) \\ Y(s) & D(s) \end{bmatrix} = \begin{bmatrix} I & 0 \\ Q(s) & I \end{bmatrix} \tag{27}$$

where $Q(s)$ is an appropriate polynomial matrix.

We see that both matrices in the product (27) are square. Therefore, by taking determinants of both sides of (27) and noting that the determinant of the matrix on the right-hand side is 1, it follows that the polynomial

$$\det_n \begin{bmatrix} X(s) & -T(s) \\ Y(s) & D(s) \end{bmatrix}$$

is independent of s. The subscript n denotes a normal determinant.

Next, (26 a) and (19) give

$$\begin{bmatrix} Es - A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} X(s) & -T(s) \\ Y(s) & D(s) \end{bmatrix} = \begin{bmatrix} I & 0 \\ Y(s) & D(s) \end{bmatrix} \tag{28}$$

Now taking the determinants of matrices on both sides of (28) and noting that the second matrix on the left-hand side is unimodular gives

$$\det_n (Es - A) \cdot c = \det_n D(s) \tag{29}$$

where c is a constant. Since no scaling is made in the realization algorithm, c must be equal to 1. Equation (29) together with the generalized degree assertion proved in the first half of the proof leads to (24). Thus the second assertion is proved.

As the final point of this section, we apply the algorithm to realize a given right MFD of the form

$$G(s) = \begin{bmatrix} s^2 + 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} s^3 + s & s + 2 \\ s^2 + s + 1 & 1 \end{bmatrix}^{-1} \tag{30}$$

Note that this is a column-pseudoproper right MFD. First decompose $D(s)$ and $N(s)$ as follows:

$$D(s) = D_{hc} S(s) + D_{lc} T(s) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c|c} s^3 & 0 \\ s^2 & 0 \\ s & 0 \\ \hline 0 & s \end{array}\right] + \left[\begin{array}{ccc|c} 0 & 1 & 0 & 2 \\ \hline 1 & 1 & 1 & 1 \end{array}\right] \left[\begin{array}{c|c} s^2 & 0 \\ s & 0 \\ 1 & 0 \\ \hline 0 & 1 \end{array}\right] \tag{31 a}$$

$$N(s) = N_c T(s) = \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] \left[\begin{array}{c|c} s^2 & 0 \\ s & 0 \\ 1 & 0 \\ \hline 0 & 1 \end{array}\right] \tag{31 b}$$

Next, construct the core realization

$$A_c = \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right], \quad B_c = \left[\begin{array}{c|c} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 1 \end{array}\right] \tag{32}$$

Finally, the realization has the following structure:

$$E = I - B_c B_c^{\mathrm{T}} + B_c D_{hc} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]$$

$$A = A_c - B_c D_{lc} = \left[\begin{array}{ccc|c} 0 & -1 & 0 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline -1 & -1 & -1 & -1 \end{array}\right]$$

$$B = B_c$$

$$C = N_c$$

Note that the formulae (15) have some nice properties which lead to certain computational advantages. For instance, for the construction of E, we simply replace the first, $(k_1 + 1)$th, ..., $(k_1 + \ldots + k_{m-1} + 1)$th rows of the identity matrix with the first, ..., the last rows of the matrix D_{hc}, and so on. A similar construction is possible for the matrix A.

5. Conclusions

A realization algorithm that determines the generalized state-space representation for a given right MFD has been presented. This algorithm does not impose conditions like column-reducedness and properness. Hence no information at infinity is lost when converting the MFD representation to its corresponding generalized state-space representation. It has also been shown that this algorithm is a natural generalization of Wolovich's (1973) algorithm; moreover, many nice properties are inherited.

As a simple algebraically dual result, a similar realization algorithm can be constructed that yields an observable generalized state-space representation from a given row-pseudoproper left MFD.

Comparing our algorithm with that in Bosgra and Van der Weiden (1981), which realizes a given polynomial system matrix into its generalized state-space representation, it appears that ours is simpler and is a more natural generalization of the existing

algorithm. Thus it can serve to bridge the conceptual gap between regular and singular systems.

There is another recently developed realization algorithm (Christodoulou and Mertzios 1985), which realizes a minimal generalized state-space representation from Markov parameters of a given transfer matrix. This algorithm is also an extension of the existing Ho–Kalman algorithm. Our algorithm together with this provides further evidence that the singular system is the more general starting point for the entire linear-system theory.

The algorithm developed here makes computer simulations possible for singular systems. It may have been observed that in simulating regular systems no differential feedback exists in the scheme; however, in simulating singular systems differential feedback does appear. By drawing a block diagram of (15), this is immediately clear.

References

Bosgra, D. H., and Van der Weiden, A. J. J., 1981, *Int. J. Control*, **33**, 393.
Christodoulou, M. A., and Mertzios, B. G., 1985, *Int. J. Control*, **42**, 1433.
Gantmacher, F. R., 1960, *The Theory of Matrices*, Vol. 1 (New York: Chelsea).
Kailath, T., 1980, *Linear Systems* (Englewood Cliffs, N.J.: Prentice-Hall).
Rosenbrock, H. H., 1974, *Int. J. Control*, **20**, 191.
Rosenbrock, H. H., and Pugh, A. C., 1974, *Int. J. Control*, **19**, 845.
Tan, S. H., and Vandewalle, J., 1986, ESAT Internal Report, K. U. Leuven, Belgium.
Wolovich, W. A., 1973, *Automatica*, **9**, 97.