

# Time-Invariant Descriptor Systems\*†

DAVID G. LUENBERGER‡

*Time-invariant descriptor systems can serve as useful models in practical situations characterized by a mixture of static and dynamic relationships. The theory is closely related to classical work on matrix pencils, but there are also important new developments.*

**Key Word Index**—Control theory; linear systems; multivariable systems; system theory; descriptor variable systems.

**Summary**—Descriptor variable systems consist of a mixture of static and dynamic equations. This paper investigates the structural characteristics of linear time-invariant descriptor systems and develops an efficient technique for converting a descriptor system to recursive form, if such a conversion is possible. The paper exploits the connection between descriptor systems and the classical theory of matrix pencils. This yields a canonical form for descriptor systems. The main contribution of the paper is the *shuffle algorithm*. This algorithm serves both as a test for the solvability of a descriptor system, and as a procedure for converting a system to recursive form, without a change of variable.

## 1. INTRODUCTION

IT IS often natural and convenient to express the equations governing a dynamic process by a system of equations of the form

$$Ex(k+1) = Ax(k) + Bu(k), \quad k=0, 1, 2, \dots, N-1. \quad (1)$$

Such a system is said to be a (discrete-time) linear time-invariant system in descriptor form. The vector  $x(k)$  is an  $n$ -dimensional descriptor vector,  $u(k)$  is an  $m$ -dimensional input vector, and  $E, A, B$  are constant matrices.  $E$  and  $A$  are  $n \times n$ , and  $B$  is  $n \times m$ . The special feature of descriptor systems is, of course, that  $E$  may be singular.

The descriptor structure arises naturally in a variety of contexts[1], but perhaps one of the most common is when the vector  $x(k)$  is composed of the variables that are the natural describing variables of the underlying system—the variables, for example, that one refers to when verbally discussing various aspects of the total system. In practice, some of the variables may be defined long before any equations are written. As a consequence, the equations (1) often include a

number of purely static equations (such as accounting identities) and, accordingly, in such situations the matrix  $E$  is singular.

Equation (1) is referred to as a time-invariant system, since the matrices  $E, A$ , and  $B$  are fixed, independent of  $k$ . The time-varying case has been treated previously[1]. As one would expect, stronger conclusions, especially concerning structure, can be deduced for the time-invariant case than for the more general case, and this paper presents these results. It should be pointed out however, that in the time-invariant case there are several alternative approaches (notably including polynomial methods[2], but see also [3] and [4]). These of course yield results that overlap with some of those presented here. The important distinction of this present work is that the fundamental concepts and the basic approach are not limited to the time-invariant case. Thus, although some of the results presented in this paper are not strictly new, one of the objectives of the paper is simply to illustrate the form of the general descriptor variable theory when specialized to the time-invariant case. The fact that in the time-invariant case the descriptor variable results are consistent with those obtainable by other procedures would appear to indicate that the general framework is perhaps a natural one.

The structural character and the behavioral pattern of a system of the form (1) can be surprisingly complex. Thus: the system may not have a solution; if it does have a solution, that solution may correspond to pure prediction of the input; and the number of degrees of freedom in the initial condition cannot always be determined by inspection. The first few sections of this paper examine the general structural properties of time-invariant descriptor systems culminating in the presentation of a canonical form.

From a practical viewpoint one is concerned primarily with those systems of form (1) which are well-behaved, and represent reasonable models of reality. Interest then turns to the development of simple procedures to test that a system

\*Received 22 August 1977; revised 16 January 1978. The original version of this paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by associate editor D. Tabak.

†The research was conducted at Systems Control, Inc., Palo Alto, and supported by the Division of Electric Energy Systems, Energy Research and Development Administration under Grant E(49-18)-2090.

‡Department of Engineering-Economic Systems, Stanford University, Stanford, California 94305, U.S.A.



the matrix  $A - Es$  be of full rank with respect to all polynomial combinations of its rows; that is, the system is solvable if and only if  $A - Es$  is not equivalent (in the sense of polynomial matrices) to a matrix with a zero row. Alternatively, (and finally) the system is solvable if and only if  $|A - Es|$  does not vanish identically.

### 3. CANONICAL STRUCTURE OF A SOLVABLE SYSTEM

Equivalence is a natural concept in the study of systems in descriptor form. Consider the time-invariant system

$$Ex(k+1) = Ax(k) + u(k) \quad (3)$$

where for simplicity the input coefficient matrix is taken to be the identity. Multiplication on the left by a nonsingular matrix  $V$  and introduction of the nonsingular change of variable  $x(k) = Wy(k)$  yields the system

$$VEWy(k+1) = VAWy(k) + v(k) \quad (4)$$

where  $v(k) = Vu(k)$  is the new vector of arbitrary inputs. The matrices  $E$  and  $A$  in the original system have been replaced by equivalent matrices  $E_1$  and  $A_1$ , each obtained by the same equivalence transformation. It is therefore quite natural to investigate the range of possible equivalent  $(E, A)$  pairs.

The study of simultaneous equivalence transformations of  $A$  and  $E$  is most conveniently investigated by consideration of the polynomial matrix  $A - Es$  referred to as a *matrix pencil*. Two pencils,  $A_1 - E_1s$  and  $A - Es$ , are equivalent if there are nonsingular matrices  $V$  and  $W$  such that  $V[A - Es]W = A_1 - E_1s$ . In this case, unlike the situation for general polynomial matrices, one requires that the matrices  $V$  and  $W$  be constant matrices. This is often emphasized by referring to this relation as *strict equivalence*. Certainly within the context of the system (3) and its alternative representation (4) attention is restricted to strict equivalence.

A matrix pencil  $A - Es$  which is square and for which  $A - Es$  does not vanish identically is traditionally termed *regular* (see for example [5]) or *nonsingular* (see for example [6]), and strong characterization results exist for this case. With either terminology, this condition precisely coincides with the concept of solvability, and hence the associated characterization results for these pencils can be directly applied.

In what follows it is convenient to refer to the *degree*  $d$  of the solvable system (3) or of the pencil  $A - Es$  as the degree of the (nonzero)

polynomial  $|A - Es|$ . Also, before stating the structure theorem itself, we consider the structure of the pure predictor, which occurs in the canonical form of a system in descriptor form.

#### The pure predictor

Consider the system (3) with

$$E = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ & & & & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & & & \ddots & \\ & & & & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

and  $k = 0, 1, 2, \dots, N$ . This system is easily verified to be solvable, for indeed  $|A - Es| = 1$ .

The corresponding individual equations are

$$\begin{aligned} x_2(k+1) &= x_1(k) + u_1(k) \\ x_3(k+1) &= x_2(k) + u_2(k) \\ &\vdots \\ 0 &= x_n(k) + u_n(k). \end{aligned} \quad (5)$$

These equations can be solved explicitly, starting with the last one, yielding

$$\begin{aligned} x_n(k) &= -u_n(k) \\ x_{n-1}(k) &= -u_n(k+1) - u_{n-1}(k) \\ &\vdots \\ x_1(k) &= -u_n(k+n-1) \\ &\quad -u_{n-1}(k+n-2) - \dots - u_1(k) \end{aligned}$$

where the equation for  $x_i(k)$  is valid for  $k = 0, 1, 2, \dots, N - n + i$ . The system represents a pure predictor, with the variable  $x_1(k)$  depending on  $u_n(k+n-1)$ . No initial conditions can be arbitrarily specified in this system. The  $n$  arbitrary constants in the solution are the *final* values of the variables.

An important special case of the general predictor system (5) is the case  $n = 1$ . This yields the scalar system

$$0 = x(k) + u(k)$$

which is a static equation without actual prediction. It is conventional to regard such a system as causal, while for any  $n > 2$  the system (5) is noncausal.

#### Structure theorem

A structure theorem for solvable systems follows directly from the classic result due to Weierstrass, see [5], on the canonical decomposition of a nonsingular matrix pencil. In the following,  $I^{(r)}$  denotes the  $r \times r$  identity matrix,  $H^{(r)}$  denotes the  $r \times r$  matrix whose elements are all zero except that those along the diagonal directly above the main diagonal are equal to 1. The matrix  $N^{(r)}$  is defined as  $N^{(r)} = I^{(r)} - H^{(r)}$ .

*Theorem 2* (Weierstrass). A nonsingular matrix pencil of degree  $d$ ,  $A - Es$ , is strictly equivalent to the pencil having the diagonal block form

$$[N^{(r_1)}, N^{(r_2)}, \dots, N^{(r_m)}; C - Is]$$

where the final block is  $d \times d$ . The integers  $r_1, r_2, \dots, r_m$  are unique, and correspond to the infinite elementary divisors of the pencil.

Of course the matrix  $C$  in the final block can be transformed by a similarity transformation to any of the standard canonical forms for square matrices. For the present purposes, however, it is not necessary to further specify  $C$ .

The system version of this theorem is the following: (for a previous systems-theoretic application of the canonical form theory of pencils to this problem see [4]).

*Theorem 2'*. A solvable system (3) of degree  $d$  is strictly equivalent to the direct sum of a number of pure predictors, purely static relations, and a system in state variable form. The dimension of the state is  $d$ .

An important special case is when each of the  $r_i$ 's in the canonical representation is 1. In this case the system is purely causal consisting of a dynamic part and a static part. Such systems are termed *regular*[1].

#### 4. THE SHUFFLE ALGORITHM

Although the canonical form derived from the classical theory of matrix pencils provides deep insight into the underlying structure of time-invariant descriptor systems, it does not always provide a convenient framework for actual computation. The main drawback is that the canonical form entails a change of variable. In most practical situations, one is usually reluctant to execute a variable change, since the original descriptor variables have contextual as well as structural significance, and since there may be additional implicit constraints, such as nonnega-

tivity constraints, on the variables. In addition, of course, the canonical form can be difficult to compute. Thus, interest turns toward the development of techniques which are computationally efficient and do not require a change of variable.

This section describes the basic shuffle algorithm as used to check solvability of a system. The extended version of the algorithm is deferred to the following section.

Solvability is a property of only the matrices  $E$  and  $A$ . Accordingly, the matrix  $B$  plays no role in the simplified version of the algorithm. The algorithm works by modifying an  $n \times (2n)$  array.

Begin with the array

$$E \quad A$$

If  $E$  is nonsingular, the procedure terminates—the system is solvable.

Otherwise, perform row operations on the whole array, bringing it to the form

$$\begin{array}{cc} T & A_1 \\ 0 & A_2 \end{array}$$

where  $T$  is of full rank. ( $T$  has  $n$  columns, but less than  $n$  rows.) The matrices  $A_1$  and  $A_2$  are a partition of the second side of the array after the row operations.  $A_1$  is the same size as  $T$ .

Next 'shuffle' the array to form

$$\begin{array}{cc} T & A_1 \\ A_2 & 0 \end{array}$$

If the  $n \times n$  matrix on the left side of the array is nonsingular, the procedure terminates—the system is solvable.

The algorithm continues in this fashion, performing row operations in order to create null rows on the left side, and then shuffling the corresponding rows from the right side to the left. The algorithm terminates in one of two ways: (1) a point is reached where the left half becomes nonsingular, in which case the system is solvable, or (2) a point is reached where there is a zero row all the way across the array, in which case the system is not solvable. The algorithm always terminates, one way or the other, in at most  $n$  steps.

*Example 1.* Starting with the  $E \ A$  array below, the shuffle progresses as indicated.

$$\begin{array}{cccccc} & E & & A & & \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{array}$$

Row operations yield

$$\begin{array}{cccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{array}$$

A shuffle yields

$$\begin{array}{cccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \end{array}$$

More row operations yield

$$\begin{array}{cccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{array}$$

Another shuffle yields

$$\begin{array}{cccc} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \end{array}$$

The algorithm terminates because the left side is nonsingular. Thus, the system is solvable.

#### Justification

An easy way to see that the shuffle algorithm checks for solvability is to consider the determinant of  $A - sE$ . According to Theorem 1, solvability is equivalent to the condition that this determinant not vanish identically.

Row operations on  $A - sE$  at most influence the determinant by a nonzero multiplicative constant. Thus, one may as well check the determinant when  $E$  has the special form obtained by the first step of the algorithm. The shuffle of  $A_2$  over to the other side of the array is equivalent to multiplication of the lower rows by  $-s$ , and each such multiplication multiplies the determinant by  $-s$ . Thus, it is clear that the shuffle algorithm is equivalent to a transformation of the original matrix pencil to a new pencil whose determinant is the original determinant multiplied by a nonzero constant and  $s^b$  where  $b$  is the total number of rows shuffled. If a point is reached where an entire row is zero, the determinant is zero. If a point is reached where the (new)  $E$  is nonsingular, the determinant is then seen to be nonzero. One of these two situations must arise within  $n$  steps, for every row shuffled increases the degree of the determinant of the

(modified) matrix pencil by one, and the maximum possible degree is  $n$ .

#### 5. THE GENERAL SHUFFLE ALGORITHM

The general shuffle algorithm accounts for the input structure of a system and produces a recursive system, equivalent to the original system. In developing the more general version, it seems best to regard the algorithm as operating directly on the original descriptor system equations (1). The general shuffle algorithm consists of the repetition of two basic operations on these equations. The first operation is that of row combination, corresponding to linearly combining individual equations. One performs such operations with the objective of obtaining an  $E$  matrix with one or more zero rows. The second operation, the shuffle, is a reindexing operation. Each (row) equation in (1) is valid for all  $k \geq 0$ , and hence  $k+1$  can be substituted for  $k$  in any row if desired. Such a substitution is used in an equation corresponding to a zero row in  $E$ . This then transfers the corresponding row in  $A$  to one in  $E$  and shifts the input terms from  $k$  to  $k+1$ . Any sequence of such row operations and time reindexing is permissible—the shuffle algorithm is a systematic procedure for obtaining a desired final form.

In the general algorithm it is often useful to restrict the kind of row operations performed, so that the final form will have a structure that is easily converted to recursive form. There are numerous variations possible, depending on the particular objectives of the situation. Two methods are outlined here.

#### Non-reduced form

The general shuffle algorithm begins with the array

$$E \quad A \quad B$$

By row operations this is brought to the form

$$\begin{array}{ccc} T & A_1 & B_1 \\ 0 & A_2 & B_2 \end{array}$$

A shuffle is performed yielding

$$\begin{array}{cccc} T & A_1 & B_1 & 0 \\ A_2 & 0 & 0 & -B_2 \end{array}$$

This corresponds to writing  $0 = A_2x(k) + B_2u(k)$  from the previous array, as  $A_2x(k+1) = -B_2u(k+1)$ . In general, any shuffle to the left of rows of  $A$  is accompanied by a shuffle to the right, and a change in sign, of all input structure elements in

the same row. The array, therefore, grows toward the right as the algorithm progresses.

When the algorithm is complete, the array will have the form

$$\bar{E} \quad \bar{A} \quad \bar{B} \quad \bar{C} \dots$$

If the system is solvable  $\bar{E}$  will be nonsingular. Thus one may write

$$x(k+1) = \bar{E}^{-1} \{ \bar{A}x(k) + \bar{B}u(k) + \bar{C}u(k+1) + \dots \} \quad (6)$$

which is a recursive structure for  $x(k)$ . This is termed a non-reduced form, since the recursion is in terms of the full descriptor vector  $x(k)$ . This is usually not the most convenient form, however, and it is slightly misleading. The vector  $x(0)$  cannot be selected arbitrarily, for there are additional equations at  $k=0$ , relating  $x(0)$  and  $u(0)$ , which were lost in the shuffle procedure. The procedure below employs a 'back shuffle' which recovers these lost equations.

#### Reduced form

The reduced form is obtained by restricting the class of row operations employed during the shuffle algorithm in order to preserve the zero rows created in  $A$ . Thus after reaching the stage

$$\begin{array}{cccc} T & A_1 & B_1 & 0 \\ A_2 & 0 & 0 & -B_2 \end{array} \quad (7)$$

rows from the upper portion are never added to the lower portion. Arbitrary row operations are permitted within each portion, and multiples of lower rows may be added to upper rows. This rule does not actually restrict the functioning of the algorithm.

Assuming the system is solvable, a final stage is reached having the form

$$\begin{array}{cccccc} T & A_1 & B_1 & C_1 & D_1 & \dots \\ A_2 & 0 & 0 & -B_2 & -C_2 & \dots \end{array}$$

The left-hand  $n \times n$  matrix can, by the allowed row operations, be brought to the special form

$$\begin{bmatrix} T \\ A_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21} & I \end{bmatrix} \quad (8)$$

which is nonsingular.\* Once this final stage is reached, the array is 'back shuffled', yielding the array

$$\begin{array}{cccc} T & A_1 & B_1 & C_1 \dots \\ 0 & A_2 & B_2 & C_2 \dots \end{array} \quad (9)$$

Using the assumed special structure for  $A_2$  and  $T$ , combinations of lower rows can be subtracted from upper rows to yield an array of the form

$$\begin{array}{cccc|cccc} I & 0 & A_{11} & 0 & B_1 & C_1 & & \\ 0 & 0 & A_{21} & I & B_2 & C_2 & \dots & \end{array} \quad (10)$$

The matrices  $B_1, C_1, \dots$  will generally have different entries than in (9).

Let  $x$  be partitioned, consistent with (10), as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Then (10) yields

$$\begin{aligned} x_1(k+1) &= A_{11}x_1(k) + B_1u(k) \\ &\quad + C_1u(k+1) + \dots \end{aligned} \quad (11a)$$

$$\begin{aligned} -x_2(k) &= A_{21}x_1(k) + B_2u(k) \\ &\quad + C_2u(k+1) + \dots \end{aligned} \quad (11b)$$

which is the reduced recursive form. The vector  $x_1(k)$  is the dynamic part, and  $x_2(k)$  is the static part of the descriptor vector. The dimension of  $x_1(k)$  is  $d$ , the degree of the system (see Section 3). Equation (11a) can be solved forward once  $x_1(0)$  is specified, although values of future inputs may be required. Equation (11b) can be solved once  $x_1(k)$  is known.

If the system is actually causal, then  $C = D = \dots = 0$  and (11) is a state vector representation of the system.

*Example 2.* Consider the  $E, A$  combination of Example 1, with input matrix

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

The row operations employed in Example 1 violate the rules that are used to obtain the reduced form, so the steps below follow a different path. The sequence of arrays is given without explanation.

\*Actually in some cases it may be necessary to permute the variables  $x_i$ ,  $i = 1, 2, \dots, n$  to obtain this form. We do not account for this possible permutation in our notation.

E			A			B		C		D	
1	0	0	0	0	1	1	0				
0	1	0	1	0	0	0	1				
0	1	0	0	1	0	0	0				
1	0	0	0	0	1	1	0				
0	1	0	1	0	0	0	1				
0	0	0	-1	1	0	0	-1				
1	0	0	0	0	1	1	0	0	0		
0	1	0	1	0	0	0	1	0	0		
-1	1	0	0	0	0	0	0	0	1		
1	0	0	0	0	1	1	0	0	0		
0	0	0	1	0	-1	-1	1	0	-1		
-1	1	0	0	0	0	0	0	0	1		
1	0	0	0	0	1	1	0	0	0	0	0
1	0	-1	0	0	0	0	0	1	-1	0	1
-1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	1	1	0	0	0	0	0
-1	1	0	0	0	0	0	0	0	1	0	0
-1	0	1	0	0	0	0	0	-1	1	0	-1

This is the final stage, which in the last step has been brought to the special form (8). The array is now back shuffled, and then brought to form (10).

1	0	0	0	0	1	1	0	0	0		
0	0	0	-1	1	0	0	-1	0	0		
0	0	0	-1	0	1	1	-1	0	1		
1	0	0	1	0	0	0	1	0	-1		
0	0	0	-1	1	0	0	-1	0	0		
0	0	0	-1	0	1	1	-1	0	1		

Thus the new representation is

$$x_1(k+1) = x_1(k) + u_2(k) - u_2(k+1)$$

$$x_2(k) = x_1(k) + u_2(k)$$

$$x_3(k) = x_1(k) - u_1(k) + u_2(k) - u_2(k) - u_2(k+1).$$

## 6. CONCLUSIONS

The descriptor variable framework is not limited to the linear time-invariant case. It has

been shown in this paper, however, that in the time-invariant case there is a complete correspondence between the concepts of solvability and conditionability and the traditional assumption of nonsingularity of the associated matrix pencil. In addition, many classical results concerning canonical structures are entirely consistent with the general descriptor variable framework. This consistency in the linear time-invariant situation indicates that the general descriptor variable results perhaps represent a natural extension of classical theory.