Boundary Recursion for Descriptor Variable Systems

DAVID G. LUENBERGER, FELLOW, IEEE

Abstract—**An n-dimensional descriptor variable system has, in general, a family of solutions. It is shown in this paper that if the system satisfies the properties of *solvability* and *conditionability*, the family of solution is n-dimensional and the set of two-point boundary values that are consistent with the system can be expressed as the solutions to a linear system of equations involving only the boundary values. This equation is termed a *boundary mapping equation* and it provides a compact representation of the system. The boundary mapping generalizes the state-transition matrix associated with state variable systems.**

If a set of n boundary conditions, independent of the boundary mapping equations, are appended to the original system, then a particular solution is defined. The boundary values can be found by solving simultaneously the boundary mapping equation, which represents the system, and the specified boundary conditions. Once the boundary values are known, the intermediate descriptor variables can be found recursively from the original system equations. Hence, the boundary mapping equation provides an effective solution method.

The boundary mapping of a descriptor variable system can itself be found by a recursive process, referred to as *boundary recursion*. In this process the boundary mappings are found over successively longer time intervals. This paper presents the boundary mapping theory, describes the recursive processes required, and applies both to some special cases, including general linear time-invariant systems and adjoint systems.

I. Introduction

Descriptor variable systems occur frequently in system theory, often as natural representations of physical or economic systems, or as the necessary conditions representing optimal control, optimal estimation, or dynamic economic equilibria. Hence, it is important to understand the structure of such systems and develop efficient methods for solving them. This paper presents a powerful and general method for solving linear descriptor variable systems based on a generalization of the state transition matrix associated with state-space systems.

A linear discrete-time descriptor variable system defined on the time interval $[0, N]$ has the form

$$E(k+1)x(k+1)=A(k)x(k)+B(k)u(k) \tag{1}$$

for $k = 0, 1, 2, \cdots, N-1$. For each k, $x(k)$ is an n-dimensional vector of *descriptor variables* and $u(k)$ is an m-dimensional vector of *input variables*. The matrices $A(k)$, $E(k+1)$, $B(k)$ are of dimensions $n \times n$, $n \times n$, and $n \times m$, respectively.

Two fundamental concepts for such systems are those of *solvability* and *conditionability* [1], and these play a central role in this paper as well. The *solvability matrix* is

$$S=\begin{bmatrix} -A(0) & E(1) & & & & \\ & -A(1) & E(2) & & & \\ & & & \ddots & & \\ & & & & E(N-1) & \\ & & & & -A(N-1) & E(N) \end{bmatrix}. \tag{2}$$

The *conditionability matrix* C is the submatrix in (2) which is enclosed by the dashed lines. The system (1) is *solvable* if S has full rank (that is, rank nN). The system (1) is *conditionable* if the matrix C has full rank [that is, rank $n(N-1)$]. It is easy to show that if either of these properties holds for the interval $[0, N]$ then it also holds on any subinterval $[k, l]$, $0 \le k < l \le N$.

Solvability means that the full system of equations represented by (1) is of full rank, so for any input sequence there is an n-dimensional linear variety of solutions. If the system is solvable, then conditionability implies that a unique solution (within the n-dimensional linear variety) can be specified by appending a total of n boundary conditions, involving the values $x(0)$ and $x(N)$. In general, for descriptor variable systems these boundary conditions cannot all be specified at one end.

It is natural to associate with (1) the homogeneous system

$$E(k+1)x(k+1)=A(k)x(k) \tag{3}$$

for $k = 0, 1, 2, \cdots, N-1$. Solvability and conditionability clearly depend only on the structure of the homogeneous system.

This system of equations defines a subspace of allowable descriptor vector sequences $x(0), x(1), \cdots, x(N)$, and this subspace will be n-dimensional if the system is solvable. We seek an efficient representation of this subspace, for then we will have an n-dimensional representation of the $n(N+1)$ unknowns that comprise the entire sequence of descriptor vectors. The key concept of this paper is that an efficient representation can be given by describing the feasible boundary pairs $(x(0), x(N))$. If the system is solvable and conditionable, a characterization of this set of boundary combinations yields a complete characterization of the system.

The system (3) puts linear restrictions on the boundary points. It is reasonable to represent such restrictions in the form

$$Z_0x(0)+Z_Nx(N)=0$$

for some matrices Z_0 and Z_N. Hence, the search for a complete characterization amounts to a search for appropriate matrices Z_0 and Z_N.

II. Boundary Mappings

Motivated by the above, we introduce the following fundamental definition.

Manuscript received October 13, 1987; revised July 8, 1988 and August 22, 1988. Paper recommended by Associate Editor, J. D. Cobb. This work was supported by the National Science Foundation under Grant NSF-ECS-8619860.

The author is with the Department of Engineering-Economic Systems, Stanford University, Stanford, CA 94305-4025.

IEEE Log Number 8825370.

0018-9286/89/0300-0287$01.00

Definition 1: A *boundary mapping* corresponding to a homogeneous descriptor variable system (3) is an $n \times 2n$ matrix $Z(0, N) = [Z_0(0, N), Z_N(0, N)]$ such that the sets of pairs $(x(0), x(N))$ satisfying

$$Z_0(0, N)x(0) + Z_N(0, N)x(N) = 0 \tag{4}$$

is exactly the same as the set of pairs $(x(0), x(N))$ for which there are intermediate values $x(1), x(2), \cdots, x(N-1)$ which together with $x(0), x(N)$ satisfy the system equations (3).

We see that in terms of characterizing boundary points, the entire system (3) can be replaced by the boundary mapping equation (4). In this sense, the boundary mapping equation is entirely equivalent to the system itself. It summarizes the restrictions placed on the end points by the system. Note, however, that the boundary mapping is not unique since it may be multiplied by a nonsingular $n \times n$ matrix.

Note also that we have carried the indexes $0, N$ to the boundary mapping Z. This is because it is useful to consider $Z(k, l)$ for various values of k, l. Indeed, the calculation of Z can be carried out recursively by changing the indexes one step at a time.

The boundary mapping is analogous to the state-transition matrix of a linear state variable system, and in fact is a direct generalization of it. To see this, consider the homogeneous state-space system

$$x(k+1) = A(k)x(k).$$

Let $\Phi(0, N)$ denote the corresponding state-transition matrix. Then the relationship $\Phi(0, N)x(0) - x(N) = 0$ holds, and fully represents the system's restrictions on the boundary points. Hence, this is the boundary mapping equation for this case, and we can set $Z(0, N) = [\Phi(0, N), -I]$.

The following theorem shows the equivalence of the boundary mapping concept and the fundamental concepts of solvability and conditionability.

Theorem 1: Corresponding to a homogeneous descriptor variable system (3), a boundary mapping $Z(k, l)$ exists and has full rank (that is, rank n) for all k, l satisfying $0 \le k < l \le N$ if and only if the system (3) is solvable and conditionable on $[0, N]$.

Proof: Let $\mathcal{B} \subset R^{2n}$ be the set of all pairs $(x(0), x(N))$ which have valid extensions as sequences $x(0), x(1), \cdots, x(N)$ that are solutions to (3). $\mathcal{B}$ is the projection of all solutions onto the $2n$-dimensional subspace represented by $(x(0), x(N))$. The existence of a boundary mapping $Z(0, N)$ of full rank is equivalent to the condition that $\mathcal{B}$ be a subspace of dimension n.

Suppose that $\dim \mathcal{B} > n$. This means that the dimension of the subspace of solutions to (3) is greater than n, which means that (3) is not solvable. Suppose that $\dim \mathcal{B} < n$. Since the dimension of the subspace of all solutions to (3) is at least n, there must in this case be two solutions with distinct sequences $x(1), x(2), \cdots, x(N-1)$ and having the same boundary values $x(0)$ and $x(N)$. This means that C is not of full rank, and hence that (3) is not conditionable. Hence, the above two arguments together show that solvability and conditionability are necessary for $\dim \mathcal{B} = n$. The converse of the two arguments establish that solvability and conditionability together are sufficient for $\dim \mathcal{B} = n$, and hence for the existence of a boundary mapping $Z(0, N)$ of full rank.

Furthermore, if (3) is solvable and conditionable over $[0, N]$, then it is also solvable and conditionable over any subinterval $[k, l]$. Hence, a $Z(k, l)$ of full rank exists. ∎

When a system is solvable and conditionable, the boundary mapping essentially collapses the system of equations (3), defined for each time period, into a single equation involving the end points. This equation completely characterizes the system since: 1) any solution to the original system satisfies the boundary mapping equation; and 2) any $(x(0), x(N))$ satisfying the boundary mapping has a *unique* extension as a solution to the complete system (3). The relation defined by the boundary mapping jumps over the intervening variables and equations while preserving their influence.

The concept of efficiently representing a descriptor variable system by its projection on the end-point space was first proposed in [2], where it was shown, in fact, that the concept works even for nonlinear systems. However, in the nonlinear case, the boundary mapping equation may itself be nonlinear. This paper applies the general concepts and prescriptions of [2] to the special case of linear systems.

III. Boundary Recursion

It was shown in [2] that boundary mappings can be computed recursively even for nonlinear systems. This implies that boundary mappings can provide an effective way to solve descriptor variable systems. The details of the recursive process for the linear case are explicitly worked out in the next few sections.

A recursive procedure for calculating the boundary mapping equation begins with a single-stage system. The very first (block) equation in the system (3) represents a boundary mapping between 0 and 1. Hence, we can define $Z(0, 1) = [-A(0), E(1)]$. We show that boundary mappings $Z(0, k)$ can be found recursively by increasing k one step at a time. Each recursion requires only elementary linear operations of dimension n.

We shall go through the first step in detail. Consider the matrix

$$\begin{bmatrix} -A(0) & E(1) & \\ & -A(1) & E(2) \end{bmatrix}. \tag{5}$$

By solvability, this matrix has full rank (that is, rank n). By conditionability, the submatrix consisting of the central (block) column also has full rank. Hence, it is possible to apply a nonsingular transformation to (5) to obtain

$$\begin{bmatrix} F & G \\ H & J \end{bmatrix} \begin{bmatrix} -A(0) & E(1) & \\ & -A(1) & E(2) \end{bmatrix} = \begin{bmatrix} Z_0 & 0 & Z_2 \\ X & I & Y \end{bmatrix} \tag{6}$$

where all submatrices are $n \times n$ and I is the identity matrix. This requires only n-dimensional linear algebra, since the essential ingredient is that of determining a basis for the central (block) column.

The upper part of (6) implies that

$$Z_0 x(0) + Z_2 x(2) = 0.$$

Since the matrix on the right of (6) is of full rank, it follows that $Z = [Z_0, Z_2]$ is of full rank. Hence, we set $Z(0, 2) \equiv Z$, and it is a boundary mapping for $[0, 2]$.

The overall matrix

$$\begin{bmatrix} X & I & Y & & & \\ Z_0 & 0 & Z_2 & & & \\ & & -A(2) & E(3) & & \\ & & & -A(3) & & \\ & & & & \ddots & \\ & & & & -A(N-1) & E(N) \end{bmatrix} \tag{7}$$

has the same rank as the original solvability matrix (since it is obtained from the original by a nonsingular transformation). Likewise the matrix (7) with the first and last (block) columns deleted has the same rank as the original conditionability matrix. Therefore, it easily follows from the position of the identity and the zero matrices in (7) that the matrix obtained by dropping the first (block) row and second (block) column

$$\begin{bmatrix} Z_0 & Z_2 & & & \\ & -A(2) & E(3) & & \\ & & -A(3) & & \\ & & & \ddots & \\ & & & -A(N-1) & E(N) \end{bmatrix} \tag{8}$$

has these same properties. This represents a homogeneous descriptor variable system which simply jumps over $x(1)$ and is defined for the variables $x(0)$, $x(2)$, $x(3)$, $\cdots$, $x(N)$. This reduced system is also solvable and conditionable. This completes this first step.

The process continues by applying an analogous transformation to the top two (block) rows and left three (block) columns of (8) to eliminate $x(2)$ and obtain $Z(0, 3)$. This process is continued until $Z(0, N)$ is obtained.

It is useful to write the recursions explicitly.

Boundary Recursion Process

Given a solvable and conditionable system (3):

1) *Initialize:* Set

$$Z_0(0, 1) = -A(0),\ Z_1(0, 1) = E(1). \tag{9}$$

2) *Recursion:* Given $Z(0, k)$, find a nonsingular $2n \times 2n$ matrix with blocks F_k, G_k, H_k, J_k to transform the (block) column $[Z_k(0, k), -A(k)]^T$ to $[0, I]^T$. Then X_k, Y_k, and $Z(0, k+1) = (Z_0(0, k+1), Z_{k+1}(0, k+1))$ are defined by

$$\begin{bmatrix} F_k & G_k \\ H_k & J_k \end{bmatrix} \begin{bmatrix} Z_0(0, k) & Z_k(0, k) & 0 \\ 0 & -A(k) & E(k+1) \end{bmatrix} = \begin{bmatrix} Z_0(0, k+1) & 0 & Z_{k+1}(0, k+1) \\ X_k & I & Y_k \end{bmatrix}. \tag{10}$$

Note that the recursion for Z can be written as

$$Z_0(0, k+1) = F_k Z_0(0, k)$$

$$Z_{k+1}(0, k+1) = G_k E(k+1). \tag{11}$$

Also, as we shall show below, the X_k's and Y_k's calculated as part of this process can be used to obtain the intermediate descriptor vectors $x(k)$, $k = 1, 2, \cdots, N-1$, once the values of $x(0)$ and $x(N)$ are known.

Example 1: Consider a simple two-period system on $[0, 2]$ with constant E and A matrices defined by

$$E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad -A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

In the time-invariant case it is known [2] that solvability and conditionability are implied by the property $\det[sE - A] \not\equiv 0$. In this case $\det[sE - A] = -s$, so the system is solvable and conditionable. Define P as the 6×6 matrix which permutes rows—interchanging 1st and 5th, 2nd and 6th. Then

$$P \begin{bmatrix} E \\ -A \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which has an identity in the bottom portion. Define R as the 6×6 matrix that carries out the row operations which zero out the top portion—subtract 5th from 2nd, and 6th from 3rd. This gives

$$RP \begin{bmatrix} -A & E & \\ & -A & E \end{bmatrix} = R \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right].$$

The upper left and right blocks define the boundary system with

$$Z_0 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad Z_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

IV. Boundary Conditions and Solution Process

The boundary mapping equation

$$Z_0(0, N)x(0) + Z_N(0, N)x(N) = 0 \tag{12}$$

completely characterizes the restrictions on $x(0)$ and $x(N)$ implied by the system (3). A particular solution to (3) is obtained by further restricting $x(0)$ and $x(N)$. This can be done by imposing an n-dimensional two-point boundary condition of the form

$$W_0 x(0) + W_N x(N) = w \tag{13}$$

where $W = [W_0, W_N]$ has rank n. However, it is clear that for (13) to represent n conditions that are independent of the system structure itself, it is necessary and sufficient that the $2n \times 2n$ matrix $[W^T, Z^T(0, N)]$ be nonsingular. This then gives a complete characterization of suitable boundary conditions for the descriptor variable system (3). We state this result as a proposition.

Proposition 1: A boundary condition (13) defines a unique solution to a solvable and conditionable system (3) if and only if the matrix $[W^T, Z(0, N)^T]$ is nonsingular.

Suppose now that a boundary condition (13) satisfying the requirements of the above proposition is specified. How can we find the corresponding sequence of descriptor vectors $x(k)$, $k = 0, 1, 2, \cdots, N$? This is done by using the matrices X_k and Y_k determined as part of the boundary recursion process.

We first use (12) and (13) together to solve for $x(0)$ and $x(N)$. We then work backward determining $x(N-1)$, $x(N-2)$, $\cdots$, $x(1)$. Specifically, assuming that $x(0)$ is known, and $x(j)$ for $k < j \leq N$ are known, then

$$x(k) = -X_k x(0) - Y_k x(k+1). \tag{14}$$

This follows directly from the decomposition (10).

V. General Solution

We now extend the analysis to the nonhomogeneous system (1). When written out for each period, this system has the form

$$\begin{bmatrix} -A(0) & E(1) & & & \\ & -A(1) & E(2) & & \\ & & \ddots & & \\ & & & -A(N-1) & E(N) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} B(0) & & & \\ & B(1) & & \\ & & \ddots & \\ & & & B(N-1) \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}. \tag{15}$$

The boundary recursion process amounts to multiplication on the left by a nonsingular matrix. If that same process is applied to (15), that nonsingular matrix will multiply the coefficient matrix on the right as well. The boundary mapping itself corresponds to one (block) row of the resulting system. Hence, taking account of the right-hand side, it is clear that the boundary mapping equation will have the form

$$Z_0(0, N)x(0)+Z_N(0, N)x(N)x(N)=v(N-1)$$

where the right-hand side $v(N-1)$ is a linear combination of the $u(k)$'s. Hence, the structure of the boundary mapping is exactly the same as the homogeneous case, except that it will have a nonzero right-hand side. In particular, this means that the Proposition concerning suitable boundary conditions of Section IV applies to the nonhomogeneous case as well.

We may determine the right-hand side as part of the boundary recursion process. It is easily seen from the top part of (6) that

$$Z_0(0, 2)x(0)+Z_2(0, 2)x(2)=v(1) \tag{16}$$

where

$$v(1)=F_1B(0)u(0)+G_1B(1)u(1).$$

It is convenient to set

$$v(0)=B(0)u(0). \tag{17}$$

Then by recursion we find

$$Z_0(0, k)x(0)+Z_k(0, k)x(k)=v(k-1) \tag{18}$$

where

$$v(k)=F_kv(k-1)+G_kB(k)u(k). \tag{19}$$

The recursion (17)–(19) therefore defines an inductive process for determining the right-hand side of the boundary mapping equation in the nonhomogeneous case.

The recursion above can be explicitly related to the matrix form of the system by noting that after k steps of the boundary recursion process the top portion of the system has the form

$$\begin{bmatrix} Z_0(0,k) & Z_k(0,k) & 0 \\ 0 & -A(k) & E(k+1)\end{bmatrix}\begin{bmatrix} x(0)\\ x(k)\\ x(k+1)\end{bmatrix} = \begin{bmatrix} v(k-1)\\ B(k)u(k)\end{bmatrix}. \tag{20}$$

It is transformed by a nonsingular transformation to

$$\begin{bmatrix} Z_0(0,k+1) & 0 & Z_{k+1}(0,k+1) \\ X_k & I & Y_k\end{bmatrix}\begin{bmatrix} x(0)\\ x(k)\\ x(k+1)\end{bmatrix} = \begin{bmatrix} F_k & G_k \\ H_k & J_k\end{bmatrix}\begin{bmatrix} v(k-1)\\ B(k)u(k)\end{bmatrix}. \tag{20}$$

Suppose now that a boundary condition of the form

$$W_0x(0)+W_Nx(N)=w \tag{22}$$

is appended to the nonhomogeneous system (1). If this boundary condition satisfies the requirements of the Proposition of Section IV, there will be a unique solution $(x(0), x(N))$ to (22) and the boundary mapping equation

$$Z_0(0, N)x(0)+Z_N(0, N)x(N)=v(N-1). \tag{23}$$

The complete sequence of descriptor vectors $x(k)$, $k = 0, 1, 2, \cdots, N$ can be found by using the X_k and Y_k matrices as in the homogeneous case and the H_k and J_k matrices. The proper recursion is easily seen from (21) to be

$$x(k)=-X_kx(0)-Y_kx(k+1)+H_kv(k-1)+J_kB(k)u(k). \tag{24}$$

Hence, starting with $x(0)$, $x(N)$ as determined from (22) and (23), the other descriptor vectors can be found in reverse order, starting with $x(N-1)$.

VI. Time-Invariant Systems

For linear time-invariant systems, the idea of generating boundary conditions that represent a system has been independently developed by others. In particular, see [3] for a good development of this idea for state-space systems, and [4] for an extension to descriptor systems. An alternative but closely related approach was developed in [5]. Of course, in the time-invariant case a much simpler recursion, based on taking powers of matrices, can be developed directly from the system description. We shall work out such a recursion here by specializing the results of the previous sections.

A time-invariant homogeneous descriptor system has the form

$$Ex(k+1)=Ax(k) \tag{25}$$

where E and A are constant, independent of k. In the case of a time-invariant system (25), the definitions of solvability and conditionability are modified to require that the solvability and conditionability matrices be of full rank for every positive integer N. It is known [6] that then solvability and conditionability are equivalent, and that each is equivalent to the requirement that det $[sE-A] \not\equiv 0$. This is in turn equivalent to the classic notion that the pencil of matrices defined by (E, A) is regular; see [7].

A system which is regular as defined above can be transformed through an equivalence transformation to a special form. Specifically, there are $n \times n$ nonsingular matrices P and Q such that [7]

$$PEQ=\begin{bmatrix} I & 0\\ 0 & E_b\end{bmatrix} \quad PAQ=\begin{bmatrix} A_f & 0\\ 0 & I\end{bmatrix} \tag{26}$$

where E_b is $n_1 \times n_1$, A_f is $n_2 \times n_2$, and $n_1 + n_2 = n$. (The subscripts b and f refer to backward and forward components of the system.)

In the classic canonic form of this type, E_b is nilpotent. However, other choices are possible. For example, it can be arranged that both E_b and A_f have all eigenvalues within the unit circle [8] (also see [5]).

We can use this representation to derive explicit recursion formulas for the boundary recursion process. The first step of that process is indicated by a partitioned form of

$$\begin{bmatrix} F & G\\ H & J\end{bmatrix}\begin{bmatrix} -A & E & 0\\ 0 & -A & E\end{bmatrix}=\begin{bmatrix} Z_0 & 0 & Z_2\\ X & I & Y\end{bmatrix}. \tag{27}$$

This is shown below. In this representation, all indicated submatrices have appropriate dimensions.

$$\left[\begin{array}{cc|cc} A_f & 0 & I & 0\\ 0 & I & 0 & E_b\\ \hline I & 0 & 0 & 0\\ 0 & 0 & 0 & -I\end{array}\right]\left[\begin{array}{cc|cc|cc} -A_f & 0 & I & 0 & 0 & 0\\ 0 & -I & 0 & E_b & 0 & 0\\ \hline 0 & 0 & -A_f & 0 & I & 0\\ 0 & 0 & 0 & -I & 0 & E_b\end{array}\right] = \left[\begin{array}{cc|cc|cc} -A_f^2 & 0 & 0 & 0 & I & 0\\ 0 & -I & 0 & 0 & 0 & E_b^2\\ \hline -A_f & 0 & I & 0 & 0 & 0\\ 0 & 0 & 0 & I & 0 & -E_b\end{array}\right]. \tag{28}$$

Hence

$$Z_0=\begin{bmatrix} -A_f^2 & 0\\ 0 & -I\end{bmatrix} \quad Z_2=\begin{bmatrix} I & 0\\ 0 & E_b^2\end{bmatrix}. \tag{29}$$

We see that the basic structural form is preserved, so that the same general block form can be used in subsequent steps.

The general solution can be seen to be

$$F_k=\begin{bmatrix} A_f & 0 \\ 0 & I \end{bmatrix} \quad G_k=\begin{bmatrix} I & 0 \\ 0 & E_b^k \end{bmatrix}$$

$$H_k=\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \quad J_k=\begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix}$$

$$Z_0(0, k)=-\begin{bmatrix} A_f^k & 0 \\ 0 & I \end{bmatrix} \quad Z_k(0, k)=\begin{bmatrix} I & 0 \\ 0 & E_b^k \end{bmatrix}. \tag{30}$$

Hence, by using these matrices in the general procedure of Section III, we have a method for solving any (regular) time-invariant system (once it has been put in the canonical form).

Suppose there are boundary conditions

$$W_0x(0)+W_Nx(N)=w. \tag{31}$$

Write the transformed version of the W matrices as

$$[W_{0f}, W_{0b}]=W_0Q^{-1}, \; [W_{Nf}, W_{Nb}]=W_NQ^{-1}.$$

Then, according to the Proposition in Section IV the boundary conditions are independent of the boundary mapping of the system if

$$\begin{bmatrix} W_{0f} & W_{0b} & W_{Nf} & W_{Nb} \\ -A_f^N & 0 & I & 0 \\ 0 & -I & 0 & E_b^N \end{bmatrix}$$

is nonsingular. Simple column operations show that this is equivalent to the requirement that the matrix

$$[W_{0f}+A_f^NW_{Nf}, \; W_{Nb}+E_b^NW_{0b}] \tag{32}$$

be nonsingular. This adds further explanation to [8] where this condition arises.

VII. Adjoint Equations

Another important and interesting example of a descriptor variable system consists of a state-space system and its adjoint. Such systems represent the necessary conditions of many optimal control problems, optimal estimation problems, and dynamic economic equilibrium problems. These equations generally have boundary conditions appended at both ends. The set of acceptable boundary conditions of such systems can be completely characterized by the boundary mapping obtained by the boundary recursion method through the Proposition of Section IV. The boundary recursion method also provides an efficient method of solution. The key observation is that the special structure of the adjoint equations is preserved by the boundary recursion process.

Consider a system

$$x(k+1)=A(k)x(k)-R(k)\lambda(k+1)$$
$$\lambda(k)=A(k)^T\lambda(k+1)+Q(k)x(k) \tag{33}$$

$k=0, 1, \cdots, N-1$. In standard descriptor variable form the system has the form

$$\begin{bmatrix} I & R(k) \\ 0 & A(k)^T \end{bmatrix}\begin{bmatrix} x(k+1) \\ \lambda(k+1) \end{bmatrix}=\begin{bmatrix} A(k) & 0 \\ -Q(k) & I \end{bmatrix}\begin{bmatrix} x(k) \\ \lambda(k) \end{bmatrix}. \tag{34}$$

Now we claim that the boundary mapping $Z(0, k)$ has a similar structure; namely,

$$\begin{bmatrix} I & B(k) \\ 0 & C(k) \end{bmatrix}\begin{bmatrix} x(k) \\ \lambda(k) \end{bmatrix}=\begin{bmatrix} D(k) & 0 \\ E(k) & I \end{bmatrix}\begin{bmatrix} x(0) \\ \lambda(0) \end{bmatrix}. \tag{35}$$

We verify this by using this assumed form to calculate $Z(0, k+1)$. We use the top of (35) to eliminate $x(k)$ in (34). This yields

$$x(k+1)=ADx(0)-AB\lambda(k)-R\lambda(k+1)$$
$$\lambda(k)=A^T(k+1)+QDx(0)-QB\lambda(k) \tag{36}$$

where we have temporarily dropped the index k on all matrices since these indexes are identical. We then solve the bottom of (36) for $\lambda(k)$ and substitute into the top of (36) and the bottom of (35). This yields

$$x(k-1)=A\{I-B[I+QB]^{-1}Q\}Dx(0)$$
$$-\{AB[I+QB]^{-1}A^T+R\}\lambda(k+1)$$
$$C[I+QB]^{-1}A^T\lambda(k+1)=\{E-C[I+QB]^{-1}QD\}x(0)+\lambda(0).$$

This has the same structure as (35) with

$$B(k+1)=AB[I+QB]^{-1}A^T+R$$
$$C(k+1)=C[I+QB]^{-1}A^T$$
$$D(k+1)=A[I+BQ]^{-1}D$$
$$E(k+1)=E-C[I+QB]^{-1}QD \tag{37}$$

where a matrix identity was used to simplify the expression for $D(k+1)$.

Suppose now that $Q(k)$ and $R(k)$ are symmetric and positive semidefinite. (This is the case in optimal control and estimation.) Then by induction we have for all k

i) $C(k)=D(k)^T$.
ii) $B(k)$ is symmetric and positive semidefinite.
iii) $[I+Q(k)B(k)]^{-1}$ and $[I+B(k)Q(k)]^{-1}$ exist.
iv) $E(k)$ is symmetric.

There is, of course, a close connection between this method for this problem and the discrete Riccati equation method of solution. The boundary recursion method is more general, and gives a complete characterization of allowable two-point boundary conditions. Furthermore, once the boundary mapping is computed, a solution to the system is easily obtained for any suitable boundary conditions.

VIII. Conclusions and Extensions

The boundary mapping is a fundamental concept for descriptor variable systems, generalizing the state transition function for state-space systems. The associated boundary recursion method is a powerful general method of solution which generalizes the recursive process for computing the transition matrix. If a descriptor variable system is solvable and conditionable, the boundary mapping is a complete summary of the system's influence on the boundary points, and the recursion method will progress without breakdown.

As pointed out in [2], the boundary recursion method for solving a system can be considered to be *dual* to the double sweep method [1] (see also [3] and [4]). The double sweep method sweeps initial conditions forward (or terminal conditions backward) so that a complete set of conditions is obtained at one end. The descriptor variables at that end can then be found, and all others found by a reverse sweep. In this method existing boundary conditions are moved closer together until they meet at a common point. The boundary recursion method on the other hand determines the boundary relations that are inherent in the system structure. Later, given boundary conditions can be appended to determine a particular solution. In this method the inherent boundary relations are moved further apart until they reach the end points of the system. This duality is an extremely important concept.

The boundary mapping theory and the boundary recursion process are presented here for linear time-varying systems. Both of these concepts, like that of the state-transition function, can be generalized to nonlinear descriptor variable systems [2]. However, as would be expected, the results are somewhat less specific than in the linear case. Nevertheless, this generalization again points out the fundamental nature of the boundary mapping concept, and it is expected that it will be useful in other investigations of descriptor variable systems.

References

[1] D. G. Luenberger, "Dynamic equations in descriptor form," *IEEE Trans. Automat. Contr.*, vol. AC-22, pp. 312–321, June 1977.

[2] ——, "Non-linear descriptor systems," *J. Economic Dynam. Contr.*, vol. 1, pp. 219–242, 1979.

[3] A. J. Krener, "Acausal realization theory, Part 1: Linear deterministic systems," *SIAM J. Contr. Optimiz.*, vol. 25, pp. 499–525, May 1987.

[4] R. Nikoukhah, A. S. Willsky, and B. C. Levy, "Boundary-value descriptor systems: Well-posedness, reachability and observability," *Int. J. Contr.*, vol. 46, pp. 1715–1737, Nov. 1987.

[5] F. L. Lewis, "Descriptor systems: Decomposition into forward and backward subsystems," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 167–170, Sept. 1984.

[6] D. G. Luenberger, "Time-invariant descriptor systems," *Automatica*, vol. 14, pp. 473–480, 1978.

[7] F. R. Gantmacher, *The Theory of Matrices*, Vol. 2. New York: Chelsea, 1960.

[8] M. B. Adams, B. C. Levy, and A. S. Willsky, "Linear smoothing for descriptor systems," in *Proc. 23rd IEEE Conf. Decision Contr.*, Las Vegas, NV, Dec. 1984, pp. 1–6.

David G. Luenberger (S'57–M'64–SM'71–F'75) was born in Los Angeles, CA, on September 16, 1937. He received the B.S. degree from the California Institute of Technology, Pasadena, in 1959 and the M.S. and Ph.D degrees from Stanford University, Stanford, CA, in 1961 and 1963, respectively, all in electrical engineering.

Since 1963, he has been on the faculty of Stanford University, where presently he is Professor and Chairman of Engineering-Economic Systems. He has developed courses and conducted research in optimization, dynamic systems, mathematical programming, and economics. He has authored three textbooks and over 60 technical publications. He has also served as consultant to several companies. His current research areas include system theory, economics, investment theory, and descriptor variable systems.

Dr. Luenberger is a member of several professional societies.