Structural Decomposition of Linear Multivariable Systems Using Symbolic Computations

Håvard Fjær Grip and Ali Saberi

Abstract—We introduce a procedure written in the mathematics software suite Maple, which transforms linear time-invariant (LTI) systems to a special coordinate basis that reveals the internal structure of the system. The procedure creates exact decompositions, based on matrices that contain elements represented by symbolic variables or exact fractions. The procedure is meant as a complement to numerical software algorithms developed by others for the same purpose. We illustrate use of the procedure by examples.

I. INTRODUCTION

In 1987 Sannuti and Saberi introduced a structural transformation of multivariable linear time-invariant (LTI) systems to a special coordinate basis (SCB) [1]. The transformation partitions a system into separate but interconnected subsystems that reflect the inner workings of the system. In particular, the SCB representation explicitly reveals the system’s finite and infinite zero structure, and invertibility properties. Since its introduction, the SCB has been used in a large body of research, on topics including loop transfer recovery, $H_2$ control, and $H_{\infty}$ control, and it has been used as a fundamental tool in the study of linear systems theory. For details, we refer to the books [2]–[6], all of which are based on the SCB, and references therein.

While the SCB provides a fine-grained decomposition of multivariable LTI systems, transforming an arbitrary system to the SCB is a complex operation. A constructive algorithm for strictly proper systems is provided in [1], based on a modified Silverman algorithm [7]. This algorithm is lengthy and involved, and includes repeated rank operations and construction of non-unique transformations to divide the state space. Thus, the algorithm can realistically be executed by hand only for very simple systems.

To automate the process of finding transformations to the SCB, numerical algorithms have been developed (see [8], [5]) and implemented as part of the Linear Systems Toolkit for Matlab [9]. Although these numerical algorithms are invaluable in practical applications, engineers often operate on systems where some or all of the elements of the system matrices have a symbolic representation. There are obvious advantages in being able to transform these systems to the SCB symbolically, without having to insert numerical values in place of symbolic variables. Furthermore, the numerical algorithms are based on inherently inaccurate floating-point operations that make them prone to numerical errors. Ideally, if the elements of the system matrices are represented by symbols and exact fractions, one would be able to obtain an exact SCB representation of that system, also represented by symbols and exact fractions. To address these issues, we have developed a procedure for symbolic transformation of multivariable LTI systems to the SCB, using the commercial mathematics software suite Maple. The procedure is based on the modified Silverman algorithm from [1], with an extension to SCB for non-strictly proper systems [10]. The purpose of this article is to introduce this procedure, and to explain how it is implemented using Maple and the LinearAlgebra package.

We believe that our procedure serves as a useful complement to available numerical tools. Symbolic transformation to the SCB makes it possible to work directly on the SCB representation of a system without first inserting numerical values, thereby removing an obstacle to more widespread use.

II. THE SPECIAL COORDINATE BASIS

In this section we give a review of the SCB. For readers unfamiliar with the topic, the complexities of the SCB may initially appear overwhelming. This is only a reflection, however, of the inherent complexities that exist in general multivariable LTI systems. For a less technical introduction to the SCB, we recommend [11]. In the following exposition, significant complexity is added to accommodate non-strictly proper systems. To get an initial overview of the SCB, we recommend ignoring the non-strictly proper case and the complexities that follow from it.

Consider the LTI system

$$\dot{x} = A\dot{x} + Bu, \quad \dot{y} = C\dot{x} + Du.$$  \hfill (1)

where $\dot{x} \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, and $\dot{y} \in \mathbb{R}^p$ is the output. We assume without loss of generality that the matrices $[B^T, D^T]^T$ and $[C, \hat{D}]$ are of full rank.

For simplicity in the non-strictly proper case (i.e., $D \neq 0$), we assume in this section that the input and output are partitioned as $\hat{u} = [u_0^T, \hat{u}_1^T]^T$, and $\hat{y} = [y_0^T, \hat{y}_1^T]^T$, where $u_0$ and $y_0$ are of dimension $m_0$, and furthermore that $\hat{D}$ has the form $\hat{D} = \text{diag}(I_{m_0}, 0)$. Then we may write

$$\hat{y} = \begin{bmatrix} y_0 \\ \hat{y}_1 \end{bmatrix} = \begin{bmatrix} C_0 \hat{x} + u_0 \\ \hat{C}_1 \hat{x} \end{bmatrix},$$ \hfill (2)
where $C_0$ consists of the upper $m_0$ rows of $\hat{C}$, and $\hat{C}_1$ consists of the remaining rows of $\hat{C}$. The special form in (2) means that the input-output map is partitioned to separate the direct-feedthrough part from the rest. Note that by substituting $\tilde{y}_0 = y_0 - C_0 \hat{x}$, we can write the system (1) in the alternative form

$$\dot{\hat{x}} = (\hat{A} - B_0 C_0) \hat{x} + \hat{B} \begin{bmatrix} y_0 \\ \hat{u}_1 \end{bmatrix}^T, \quad \hat{y} = \hat{C} \hat{x} + \hat{D} \hat{u},$$

(3)

where $B_0$ consists of the left $m_0$ columns of $\hat{B}$. In the strictly proper case, $B_0$ and $C_0$ are nonexistent.

By nonsingular transformation of the state, output, and input, the system (1) can be transformed to the SCB. We use the symbols $x$, $y$, and $u$ to denote the state, output, and input of the system transformed to SCB form. The transformations between the original system (1) and the SCB are called $\Gamma_1$, $\Gamma_2$, and $\Gamma_3$, and we write $\hat{x} = \Gamma_1 x$, $\hat{y} = \Gamma_2 y$, and $\hat{u} = \Gamma_3 u$.

The state $x$ is partitioned as $x = \text{col}(x_a, x_b, x_c, x_d)$, where each component represents a particular subsystem described in the next section. The output is partitioned as $y = \text{col}(y_0, y_d, y_b)$, where $y_0$ is the original output $y_0$ from (1), $y_d$ is the output from the $x_d$ subsystem, and $y_b$ is the output from the $x_b$ subsystem. The input is partitioned as $u = \text{col}(u_0, u_d, u_c)$, where $u_0$ is the original input $u_0$ from (1), $u_d$ is the input to the $x_d$ subsystem, and $u_c$ is the input to the $x_c$ subsystem. Because $u_0$ appears first in both $\hat{u}$ and $u$, $\Gamma_3$ is on the form $\text{diag}(I_{m_0}, \hat{\Gamma}_3)$, for some nonsingular $\hat{\Gamma}_3$.

A. Structure of the SCB

Consider first the case when (1) is strictly proper. The meaning of the four subsystems can be explained as follows:

* The $x_a$ subsystem represents the zero dynamics. This part of the system is not directly affected by any inputs, nor does it affect any outputs directly. It may be affected, however, by the outputs $y_b$ and $y_d$ from $x_b$ and $x_d$ subsystems.

* The $x_b$ subsystem has a direct effect on the output $y_b$, but it is not directly affected by any inputs. It may be affected, however, by the output $y_d$ from the $x_d$ subsystem. The $x_b$ subsystem is observable from $y_b$.

* The $x_c$ subsystem is directly affected by the input $u_c$, but it does not have a direct affect on any outputs. It may also be affected by the outputs $y_b$ and $y_d$ from the $x_b$ and $x_d$ subsystems, as well as the state $x_a$. The $x_c$ subsystem is controllable from $u_c$.

* The $x_d$ subsystem represents the infinite zero structure. This part of the system is directly affected by the input $u_d$, and it also affects the output $y_d$ directly. The $x_d$ subsystem can be further partitioned into $m_d$ single-input single-output (SISO) subsystems $x_i$ for $i = 1, \ldots, m_d$. Each of these subsystems consist of a chain of integrators of length $q_i$ from the $i$’th element of $u_d$ to the $i$’th element of $y_d$. Each integrator chain may be affected at each stage by the output $y_d$ from the $x_d$ subsystem, and at the lowest level of the integrator chain (where the input appears), it may be affected by all the states of the system. The $x_d$ subsystem is observable from $y_d$ and controllable from $u_d$.

For non-strictly proper systems the structure is the same, except for the existence of the direct-feedthrough output $y_0$, which is directly affected by the input $u_0$, and can be affected by any of the states of the system. It can also affect all the states of the system.

B. SCB Equations

The SCB representation of the system (1) is given by

$$\dot{x}_a = A_{aa} x_a + B_{ab} u_0 + L_{ad} y_d + L_{ah} y_b, \quad (4a)$$

$$\dot{x}_b = A_{bb} x_b + B_{bh} u_0 + L_{bd} y_d, \quad (4b)$$

$$x_c = A_{cc} x_c + B_{co} u_0 + L_{cd} y_d + L_{cb} y_b + A_{ca} x_a + B_{cu} u_c, \quad (4c)$$

$$\dot{x}_i = \tilde{A}_i x_i + \tilde{B}_i u_0 + L_{id} y_d + B_{di} (u_i + \tilde{E}_{ia} x_a + \tilde{E}_{ib} y_b + \tilde{E}_{ic} x_c + \tilde{E}_{id} y_d), \quad (4d)$$

where $i = 1, \ldots, m_d$. The outputs are given by

$$y_0 = C_{0a} x_a + C_{0b} x_b + C_{0c} x_c + C_{0d} x_d + u_0, \quad (5a)$$

$$y_i = C_{qi} x_i, \quad i = 1, \ldots, m_d, \quad (5b)$$

$$y_b = C_{ib} x_b. \quad (5c)$$

The $q_i$-dimensional states $x_i$ make up the state $x_d = \text{col}(x_1, \ldots, x_{m_d})$; the scalar outputs $y_i$ make up the output $y_d = \text{col}(y_1, \ldots, y_{m_d})$; and the scalar inputs $u_i$ make up the input $u_d = \text{col}(u_1, \ldots, u_{m_d})$. The matrices $A_{qi}$, $B_{qi}$, and $C_{qi}$ have the special structure

$$A_{qi} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B_{qi} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \quad C_{qi} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}. \quad (5d)$$

The pair $(A_{bb}, C_{ib})$ is observable, and the pair $(A_{cc}, B_{ci})$ is controllable. In the strictly proper case, the input $u_0$ and output $y_0$ are nonexistent, as are the matrices $B_{db}$, $B_{bh}$, $B_{co}$, $B_{db}$, $C_{0a}$, $C_{0b}$, $C_{0c}$, and $C_{0d}$.

C. Compact Form

We may write (4) as

$$\dot{x} = Ax + B \begin{bmatrix} y_0 \\ u_0^T \\ u_c^T \end{bmatrix}^T, \quad y = Cx + Du,$$

(6)

with the SCB system matrices $A$, $B$, $C$, and $D$ defined as

$$A = \begin{bmatrix} A_{aa} & L_{ab} C_b & 0 & L_{ad} C_d \\ 0 & A_{bb} & 0 & L_{bd} C_d \\ A_{ca} & L_{cb} C_b & A_{cc} & L_{cd} C_d \\ B_d E_{da} & B_d E_{db} & B_d E_{dc} & A_{dd} \end{bmatrix}, \quad B = \begin{bmatrix} B_{d0} & 0 \\ 0 & B_{d0} & 0 \\ 0 & B_{c0} & B_c \end{bmatrix},$$

$$C = \begin{bmatrix} C_{0a} & C_{0b} & C_{0c} & C_{0d} \\ 0 & 0 & 0 & C_d \\ 0 & C_b & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} I_{m_0} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where $A_{dd} = \text{diag}(A_{q_1}, \ldots, A_{q_{m_d}}) + L_{dd} C_d + B_d E_{dd}$, $B_d = \text{diag}(B_{q_1}, \ldots, B_{q_{m_d}})$, $C_d = \text{diag}(C_{q_1}, \ldots, C_{q_{m_d}})$, $L_{dd} = [L_{1d}^T, \ldots, L_{m_d}^T]^T$, $E_{dd} = [E_{d1}^T, \ldots, E_{dm_d}^T]^T$, and similar for $E_{db}$, $E_{dc}$, and $E_{dd}$.

To see the relationship between the system matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$ from (6) and the SCB matrices $A$, $B$, $C$, and $D$, substitute $\hat{x} = \Gamma_1 x$, $\hat{y} = \Gamma_2 y$, and $\hat{u} = \Gamma_3 u$ in equation (3). Also, note that since $\Gamma_3$ is of the form $\text{diag}(I_{m_0}, \hat{\Gamma}_3)$, we can make the substitution $\text{col}(y_0, \hat{u}_1) = \Gamma_3 \text{col}(y_0, u_0, u_c)$. We then obtain the equations

$$\dot{x} = \Gamma_1^{-1} (\hat{A} - B_0 C_0) \Gamma_1 x + \Gamma_1^{-1} \hat{B} \Gamma_3 \begin{bmatrix} y_0 \\ u_0^T \\ u_c^T \end{bmatrix}^T,$$
\[ y = \Gamma_2^{-1} \hat{C} \hat{1} \hat{x} + \Gamma_2^{-1} \hat{D} \Gamma_3 u. \]

Comparison with (6) then shows that \( A = \Gamma_1^{-1} ( \hat{A} - B \hat{C}_0 ) \Gamma_1, \)
\( B = \Gamma_1^{-1} \hat{B} \Gamma_3, \)
\( C = \Gamma_2^{-1} \hat{C} \Gamma_1, \)
and \( D = \Gamma_2^{-1} \hat{D} \Gamma_3 \). In the strictly proper case, the expression for \( A \) reduces to \( A = \Gamma_1^{-1} \hat{A} \Gamma_1 \).

D. Pre-Transformation of Non-Strictly Proper Systems

We assumed initially that the input and output vectors \( \hat{u} \) and \( \hat{y} \) have a special partitioning that separates the direct-feedthrough part from the rest, as shown in (2). A strictly proper system already has this form, but given a general non-strictly proper system, a pre-transformation may have to be applied to put the system in the required form. Suppose that the we initially have a system with input \( \hat{u} \), output \( \hat{y} \), input matrix \( \hat{B} \), and output matrices \( \hat{C} \) and \( \hat{D} \). Then there are nonsingular transformations \( U \) and \( Y \) such that \( \hat{u} = U \tilde{u} \) and \( \hat{y} = Y \tilde{y} \), where \( \tilde{u} \) and \( \tilde{y} \) have the structure required in (2). The dimension \( m_0 \) of \( \tilde{u}_0 \) and \( \tilde{y}_0 \) is the rank of \( \hat{D} \). The matrices \( \hat{B}, \hat{C}, \) and \( \hat{D} \) are obtained from \( \hat{B}, \hat{C}, \) and \( \hat{D} \) by \( \hat{B} = BU, \hat{C} = Y^{-1} \hat{C}, \) and \( \hat{D} = Y^{-1} \hat{D}U \). Our Maple procedure, in addition to returning the matrices \( A, B, C, \) and \( D \) of the SCB system, the transformations \( \Gamma_1, \Gamma_2, \) and \( \Gamma_3 \) to transform (1) to SCB form, and the dimension of each subsystem, returns the transformations \( U \) and \( Y \), to take a general non-strictly proper system to the form required of (1).

E. Some Properties of the SCB

Some simple properties of the SCB that we shall refer to later are the following: (i) The invariant zeros of the system (1) are the eigenvalues of the matrix \( A_{uu} \). Hence, the system is minimum-phase if, and only if, the eigenvalues of \( A_{uu} \) are located in the open left-half complex plane. (ii) The system (1) is right-invertible if, and only if, the subsystem \( x_b \) is nonexistent. (iii) The system (1) is left-invertible if, and only if, the subsystem \( x_b \) is nonexistent. For a much more detailed treatment of the properties of the SCB, see [6, Ch. 3].

III. MAPLE PROCEDURE

Our Maple procedure is invoked as follows:

\[ A, B, C, D, G_1, G_2, G_3, U, Y, \dim := \text{scbSP}(A, B, C, D); \]

The inputs \( A_i, B_i, C_i, \) and \( D_i \) are system matrices describing a general multivariable \textit{LTI} system. The outputs \( A, B, C, \) and \( D \) are the system matrices describing a corresponding SCB system. The outputs \( G_1, G_2, \) and \( G_3 \) are the transformation matrices \( \Gamma_1, \Gamma_2, \) and \( \Gamma_3 \) between the system (1) and the SCB. The outputs \( U \) and \( Y \) are the pre-transformations that must be applied to the system to put it in the form required of (1), as described in Section II-D. Finally, the output \( \dim \) is a list of four integers representing the dimensions of the \( x_q, x_b, x_s, \) and \( x_d \) subsystems, in that order. The Maple source code is available from [12].

The modified Silverman algorithm for transformation to the SCB is much too long to be presented in this article. For the details of the algorithm, we refer to [1]. In the following we shall present a broad outline of the steps of the algorithm, and discuss issues that require particular attention in a symbolic implementation. Much of the algorithm consists of tedious but straightforward manipulation of matrices, which is not discussed in this article.

Throughout the algorithm, we identify a large number of variables that are linear transformations of the original state. We keep track of these by storing the matrices that transform the original state to the new variables. For example, the temporary variable \( y_0 \), given by the expression \( y_0 = C \hat{x} \), is represented internally by a \textit{Matrix} data structure containing \( C \). The procedure is not written to perform well on floating-point data. For this reason, all floating-point elements of the matrices passed to the procedure are converted to exact fractions before any other operations are performed, using Maple’s \texttt{convert} function.

A. Strictly Proper Case

The algorithm for strictly proper systems is implemented as \texttt{scbSP}. The first part of this algorithm identifies the two subsystems that directly influence the outputs, namely the \( x_b \) and \( x_d \) subsystems, through a series of steps that are repeated until the outputs are exhausted. The algorithm works by identifying transformed input and output spaces such that each input channel is directly connected to one output channel by a specific number of inherent integrations.

Let the strictly proper system passed to the \texttt{scbSP} procedure be represented by the state equations \( \dot{x} = A \hat{x} + B \hat{u}, \dot{y} = C \hat{x} \). In the first iteration we start with the output \( y_{10} = \hat{C} \hat{x} \) and determine whether its derivative \( \dot{y}_{10} = \hat{C} \dot{\hat{x}} + \hat{C} \hat{B} \hat{u} \) depends on any part of the input \( \hat{u} \). If so, we use a transformation \( S_1 \) to separate out a linear combination of outputs and inputs that are separated by one integration in a linearly independent manner. This will create an integrator chain of length one, as part of the \( x_d \) subsystem. A transformed part of the output derivative that is not directly influenced by the input is denoted \( \hat{C}_1 \hat{x} \), and is processed further. We use a transformation \( \hat{C}_1 \hat{x} \) to separate out any part of \( \hat{C}_1 \hat{x} \) that is linearly dependent on \( y_{10} \). This will create states that are part of the \( x_b \) subsystem. After the linearly dependent components are separated out, the remaining part of the output derivative is given the name \( y_{20} \). In the next iteration we process \( y_{20} \) in the same fashion as \( y_{10} \), to identify integrator chains of length two, and possibly further additions to the \( x_b \) subsystem. The algorithm continues in this fashion until the outputs are exhausted.

1) Constructing Transformation Matrices: When implementing these steps in Maple, the main part of each iteration consists of constructing transformation matrices \( S_i \) and \( \phi_i \). In particular, we are faced with the following problem at step \( i \): given a matrix \( C_i \) of dimension \( p_i \times n \) and a matrix \( D_{i-1} \) of dimension \( q_{i-1} \times m \) of maximal rank \( q_{i-1} \), find a nonsingular matrix \( S_i \) such that

\[ S_i \begin{bmatrix} D_{i-1} \\ C_i \hat{B} \end{bmatrix} = \begin{bmatrix} \hat{D}_{i-1} \\ \hat{C}_i \hat{B} \end{bmatrix}, \]

where \( \hat{D}_i \) is a \( q_i \times m \) matrix of maximal rank, and where \( S_{i1}, S_{i2}, \) and \( S_{ib} \) are of dimensions \( q_i \times p_i, (p_i - q_i) \times p_i, \) and
the state of the original system, it is generally desirable helps avoid unnecessary changes to the original states, and therefore construct it generally produces more appealing solutions. We require that the rows of \( \bar{S}_b \) and \( \bar{S}_d \) must be linearly independent of the already identified states \( x_b \) and \( x_d \), so that \( x_s, x_b, \) and \( x_d \) together span the entire state space; and that its derivative \( \dot{x}_s \) must only depend on \( x_s \) itself, plus \( y_b, y_d, \) and \( u_c, \) because those are the only quantities allowed in the derivatives of \( x_d \) and \( x_s \) in the strictly proper case.

Suppose that \( \text{col}(x_b, x_d) = \Gamma_{bd} \). The procedure for finding \( x_s \) is to start with a temporary state vector \( x_s^0 = \Gamma_{bd} x_b \) that is linearly independent of \( x_b \) and \( x_d \). Hence, we select \( \Gamma_5^0 \) such that \( [\Gamma_5^0, \Gamma_{bd} x_b] \) is nonsingular. To do so in our Maple procedure, we use the same technique as for finding \( S_1 \) based on \( S_2 \) in Section III-A.1.

The derivative of \( x_s^0 \), written in terms of the states \( x_s^0, x_b, \) and \( x_d \), and the inputs \( u_c \) and \( u_{id} \), can be written as

\[
x_s^0 = A^0 \begin{bmatrix} x_s^0 & x_b & x_d \end{bmatrix}^T + B^0 \begin{bmatrix} u_c \end{bmatrix}^T
\]

\[
= A^0 \hat{x}_s^0 + A_{sb}^0 x_b + A_{sd}^0 x_d + B_{sc}^0 u_c + B_{sd}^0 u_{id},
\]

for some matrices \( A^0 = [A_{sb}^0, A_{sd}^0, A_{sc}^0] \) and \( B^0 = [B_{sc}^0, B_{sd}^0] \). In our Maple procedure, we can easily calculate \( A^0 = \Gamma_5^0 \Gamma_{bd}^{-1} x_b \) and \( B^0 = \Gamma_5^0 \Gamma_{bd}^{-1} x_d \), and then extract the matrices \( A_{sb}^0, A_{sd}^0, A_{sc}^0, B_{sc}^0, \) and \( B_{sd}^0 \). To do so, we use the \texttt{MatrixInverse} procedure of the Maple Algebra package.

To conform with the SCB, we need to modify \( x_s^0 \) to eliminate the input \( u_{id} \) in \( x_s^0 \). To eliminate \( u_{id} \), we create a temporary state vector \( x_{s0} = \Gamma_{bd} x_b \), consisting of the lowermost level of each integrator chain in the \( x_d \) subsystem (that is, the point where the input enters the integrator chain). According to (4), we then have \( x_{s0} = u_d + A_{d0} \begin{bmatrix} x_s^0 & x_b \end{bmatrix}^T \), for some matrix \( A_{d0} \). Therefore, by defining a new temporary state \( x_s^1 = x_s^0 - B_{sc}^0 u_c \), we have \( x_s^1 = \hat{A}_s \begin{bmatrix} x_s^0 & x_b \end{bmatrix}^T + B_{sc}^0 u_c \). Hence, the derivative of the new temporary state vector \( x_s^1 \) is independent of \( u_{id} \), bringing us one step closer to obtaining \( x_s \). The elimination procedure is continued in a similar fashion, as described in [1], until we obtain a state \( x_s \) such that \( x_s \) depends only on \( x_s, y_b, y_d, \) and \( u_c \).

The final step is to decompose \( x_s \) into a subsystem \( x_d \) that is unaffected by the input \( u_c \), and a subsystem \( x_s \) that is controllable from \( u_c \). We do this by transforming \( x_s \) to the Kalman controllable canonical form. First, we find the derivative \( \dot{x}_s = A_{ss} x_s + A_{sp} y_b + A_{sd} x_d + B_{sc} u_c \), for some matrices \( A_{ss}, A_{sp}, A_{sd}, \) and \( B_{sc} \). We then compute the controllability matrix \( C_{ct} = [B_{sc}, A_{ss} B_{sc}, \ldots, A_{ss}^{n-1} B_{sc}] \), where \( n_s \) is the dimension of \( x_s \). To transform \( x_s \) to the Kalman controllable canonical form, we define \( \phi(x_s, x_c) = [T, T_1]^{-1} x_s \). 

\[ (p_l - q_i) \times \bar{q}_{i-1} \]. The meaning of the various dimensions are not important in this context. In general, \( \bar{S}_i \) is not unique.

The rank of the matrix \( \hat{D}_{T-1}^{T}, (C_b)^T \) can be obtained with the \texttt{Rank} procedure in the LinearAlgebra package. To construct the matrix \( \bar{S}_i \), the first observation we make is that, since \( S_b \hat{D}_{T-1} + S_b C_b = 0 \), the rows of the matrix \( \hat{S}_{ib}, \hat{S}_{2} \) must belong to the left null space of \( \hat{D}_{T-1}^{T}, (C_b)^T \). If \( \hat{D}_{T-1}^{T}, (C_b)^T \) has full rank \( \bar{q}_{i-1} - p_l, \) then \( S_{ib} \) and \( S_2 \) are empty matrices, and we may select \( S_{ia} = 0 \) and \( S_{1} = I \). Otherwise, we can obtain a set of linearly independent basis vectors for the left null space of \( \hat{D}_{T-1}^{T}, (C_b)^T \), or equivalently, for the right null space of its transpose, using the \texttt{NullSpace} procedure of the LinearAlgebra package. The transpose of the basis vectors can then be stacked to form the matrix \( [S_{ib}, S_2] \), which can be split up to form \( \bar{S}_b \) and \( \bar{S}_2 \). However, the null space basis is not unique and, moreover, the order in which the basis vectors are returned by Maple is not consistent. This may cause our procedure to produce different results on different executions with the same matrices, which is undesirable. To avoid this, we first stack the transpose of the basis vectors, and then transform the resulting matrix to the unique reduced-row echelon form, by using the \texttt{ReducedRowEchelonForm} procedure of the LinearAlgebra package. Since the transformation involves a finite number of row operations, the rows of the matrix in reduced-row echelon form remain in the left null space.

Since \( \bar{S}_i \) should be a nonsingular matrix, the submatrix \( S_i \) must be nonsingular. This requires that \( S_2 \) has maximal rank, which is confirmed as follows: if any of the rows of \( S_2 \) are linearly dependent, a linear combination of rows in \( S_{ib}, S_2 \) cannot be constructed to create a row vector \( v \) such that \( v^T \hat{D}_{T-1}^{T}, (C_b)^T = 0 \), where the rightmost \( p_l \) columns of \( v \) are zero. However, since the rows of \( \hat{D}_{T-1} \) are linearly independent, this implies that \( v = 0 \), which in turn implies that \( S_{ib}, S_2 \) must have linearly dependent rows. Since this is not the case, \( S_2 \) must have maximal rank.

We continue by constructing the matrix \( S_{1} \). Nonsingularity of \( S_i \) requires that the rows of \( S_1 \) must be linearly independent of the rows of \( S_2 \). One way to produce \( S_1 \) is to choose its rows to be orthogonal to the rows of \( S_2 \), which can be achieved by using a basis for the right null space of \( S_2 \). However, since the matrix \( S_i \) will be used to transform the state of the original system, it is generally desirable for this matrix to have the simplest possible structure. This helps avoid unnecessary changes to the original states, and thus it generally produces more appealing solutions. We therefore construct \( S_1 \) by the following procedure: we start by initialising \( S_1 \) as the identity matrix of dimension \( p_l \times p_l \). We then create a reduced-row echelon form of \( S_2 \), and iterate backwards over the rows of this matrix. For each row, we search along the columns from the left until we reach the leading 1 on that row. We then delete the row in \( S_1 \) corresponding to the column with the leading 1. This ensures that \( S_i = [S_{1}^T, S_{2}^T] \) is nonsingular, with \( S_i \) consisting of zeros except for a single element equal to 1 on each row. The construction of \( S_i \) is now easily completed.

At each step, we must also construct a nonsingular matrix \( \phi \). The problem of finding this matrix is analogous to the problem of finding \( S_i \), and we therefore use the same procedure. Finding the transformations \( S_i \) and \( \phi \) constitute the most important part of finding the states \( x_b \) and \( x_d \). After \( x_b \) and \( x_d \) are identified, finding the input and output transformations \( \Gamma_3 \) and \( \Gamma_2 \) is straightforward, based on [1].
where the columns of $T_1$ span the column space of $C_{ctr}$ and the columns of $T_2$ are orthogonal to the columns of $T_1$. We obtain a set of linearly independent basis vectors for the column space of $C_{ctr}$ using the `ColumnSpace` procedure from the LinearAlgebra package. To create $T_1$, we first create a matrix by stacking the transpose of the basis vectors. $T_2$ is then chosen as the transpose of the reduced-row echelon form of that matrix. We create $T_2$ in a similar fashion, based on a linearly independent set of basis vectors for the left null space of $C_{ctr}$, which is orthogonal to the column space of $C_{ctr}$. We can now compute the transformation matrix $\Gamma_1$.

C. Non-Strictly Proper Case

To handle the non-strictly proper case, the first step is to find the pre-transformation matrices $U$ and $Y$, described in Section II-D. Suppose that the matrices passed to the procedure `scb` are $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$. We need to find nonsingular $U$ and $Y$ such that, according to Section II-D, $\hat{B} = BU$, $\hat{C} = Y^{-1}\hat{C}$, and $\hat{D} = Y^{-1}\hat{D}U$, where $\hat{D}$ is of the form $\text{diag}(I_{m_0}, 0)$. The rank $m_0$ of $\hat{D}$ is found using the `Rank` procedure. Let $Y^{-1} = [Y_1^T, Y_2^T]^T$, where $Y_1$ has $m_0$ rows. Then we have the equations $Y^{-1}\hat{D}U = \{ (Y_1\hat{D}U)^T, (Y_2\hat{D}U)^T \}^T$, where $Y_1\hat{D}U = [I_{m_0}, 0]$ and $Y_2\hat{D}U = 0$. To solve these equations, we choose the rows of $Y_2$ from the left null space of $\hat{D}$, using `NullSpace` and `ReducedRowEchelonForm` as before; and we select $Y_1$ such that $[Y_1^T, Y_2^T]^T$ is nonsingular, using the same procedure as for finding $S_{11}$ given $S_2$ in Section II-A.1. This leaves us to solve the equation $Y_1\hat{D}U = [I_{m_0}, 0]$ with respect to some nonsingular $U$. Let $U^{-1} = [U_1^T, U_2^T]$ such that $U_1$ has $m_0$ rows. We select $U_1 = Y_1\hat{D}$, and we select $U_2$ such that $[U_1^T, U_2^T]^T$ is nonsingular, by the same procedure as before. It is then straightforward to confirm that $Y_1\hat{D}U = [I_{m_0}, 0]$. We can now calculate the matrices $\hat{B}$, $\hat{C}$, and $\hat{D}$ that conform with the required structure of (1).

Let $B_0$ consist of the left $m_0$ columns of $\hat{B}$, and $\hat{B}_1$ consist of the remaining columns of $\hat{B}$. Similar to (3), we can write the system equations (1) as

\[
\dot{x} = (\hat{A} - B_0C_0)x + B_0y_0 + \hat{B}_1\hat{u}_1,
\]

\[
y_0 = C_0x + u_0,
\]

\[
\hat{y}_1 = \hat{C}_1\hat{x},
\]

Suppose we obtain the SCB form of the strictly proper system described by the matrices $(\hat{A} - B_0C_0)$, $\hat{B}_1$, and $\hat{C}_1$, by invoking the procedure `scbSP`, and suppose the transformation matrices returned for this system are $\Gamma_1$, $\Gamma_2$, and $\Gamma_3$. Substituting $\hat{x} = \Gamma_1x$, $\hat{y}_1 = \Gamma_2[y_1^T, y_2^T]^T$, and $\hat{u}_1 = \Gamma_3[u_1^T, u_2^T]^T$ in (7) yields

\[
\dot{\hat{x}} = \Gamma^{-1}_1(\hat{A} - B_0C_0)\hat{x} + \Gamma^{-1}_1B_0y_0 + \Gamma^{-1}_1\hat{B}_1\Gamma_3[u_1^T, u_2^T]^T,
\]

\[
y_0 = C_0\Gamma_1x + u_0,
\]

\[
[y_1^T, y_2^T]^T = \Gamma^{-1}_2\hat{C}_1\Gamma_1x.
\]

It is easily confirmed that this system conforms to the SCB, by defining $A = \Gamma^{-1}_1(\hat{A} - B_0C_0)\Gamma_1$, $B = \Gamma^{-1}_1B_1\Gamma_3$, $C = \Gamma^{-1}_2\Gamma_3^T\Gamma_1$, and $D = \text{diag}(I_{m_0}, 0)$. Defining the transformation matrices for the non-strictly proper system as $\Gamma_1 = \Gamma_1$, $\Gamma_2 = \text{diag}(I_{m_0}, \Gamma_2)$, and $\Gamma_3 = \text{diag}(I_{m_0}, \Gamma_3)$, we obtain $A = \Gamma^{-1}_1(\hat{A} - B_0C_0)\Gamma_1$, $B = \Gamma^{-1}_1\hat{B}_1\Gamma_3$, $C = \Gamma^{-1}_2\hat{C}_1\Gamma_1$, and $D = \Gamma^{-1}_2D\Gamma_3$, which are the proper expressions relating the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$ to the SCB matrices (see Section II-C).

IV. EXAMPLE: LINEAR SINGLE-TRACK MODEL

A widely used model for the lateral dynamics of a car is the linear single-track model (see, e.g., [13]). For a car on a horizontal surface, this model is described by the equations $\dot{v}_y = \frac{1}{m}(F_l + F_r) - rv_x$, $\dot{r} = \frac{1}{J}(l_iF_l - l_iF_r)$, where $v_y$ is the lateral velocity at the center of gravity; $r$ is the yaw rate (angular rate around the vertical axis); $m$ is the mass; $J$ is the moment of inertia; $l_i$ and $l_r$ are the longitudinal distances from the center of gravity to the front and rear axles; and $F_l$ and $F_r$ are the lateral road-tire friction forces on the front and rear axles. The longitudinal velocity $v_x$ is assumed to be positive and to vary slowly enough compared to the lateral dynamics that it can be considered a constant. The friction forces can be modeled by the equations $F_l = \frac{c_l}{v_y + l_i}\left(\delta_l - \frac{v_x}{v_y} - l_i\right) - \frac{1}{J}F_l$ and $F_r = \frac{c_r}{v_y + l_i}\left(-\delta_r + \frac{v_x}{v_y} - l_i\right) - \frac{1}{J}F_r$, where $\delta_l$ is the front-axle steering angle; $c_l$ and $c_r$ are the front- and rear-axle cornering stiffnesses; and $T_i$ is a speed-dependent tire relaxation constant (see, e.g., [14]). In modern cars with electronic stability control, the main measurements that describe the lateral dynamics are the yaw rate $r$ and the lateral acceleration $a_y = \frac{1}{m}(F_l + F_r)$. Considering $\delta_l$ as the input, the system is described by

\[
\hat{A} = \begin{bmatrix}
0 & -v_x & \frac{1}{m} & \frac{1}{m} \\
0 & 0 & \frac{1}{v_y} & -\frac{1}{v_y} \\
-c_r & -\frac{c_r}{v_y} & -\frac{1}{v_y} & 0 \\
-c_l & -\frac{c_l}{v_y} & 0 & -\frac{1}{v_y}
\end{bmatrix},
\]

\[
\hat{B} = \begin{bmatrix}
0 \\
0 \\
\frac{1}{v_y} \\
0
\end{bmatrix},
\]

\[
\hat{C} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & \frac{1}{m}
\end{bmatrix},
\]

\[
\hat{D} = \begin{bmatrix}
0 \\
0
\end{bmatrix}.
\]

If we pass these matrices to our Maple procedure, we obtain SCB system matrices

\[
A = \begin{bmatrix}
-\frac{1}{v_i} & 1 & 0 & 0 & \frac{T_{el,m}}{c_r(l_i + \bar{c}_r)} \\
-\frac{c_r}{v_i} & 0 & 1 & 0 & \frac{T_{el,m}}{c_r(l_i + \bar{c}_r)} \\
-c_l & 0 & 0 & 0 & \frac{1}{v_i} \\
-c_l & 0 & 0 & 0 & \frac{1}{v_i}
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

\[
D = \begin{bmatrix}
0 \\
0
\end{bmatrix}.
\]

The dimension list $\text{dim}$ returned by the procedure is 0, 3, 0, 1, meaning that the first three states belong to the $x_d$ subsystem, and the last state is an integrator chain of length 1, belonging to the $x_d$ subsystem. Inspection of the SCB matrices immediately reveals that the system is observable, since both the $x_d$ and $x_d$ subsystems are always observable. The system is left-invertible, since the state $x_d$ is non-existent, meaning that the steering angle can be identified from the outputs. The system is not right-invertible, since it has an $x_d$ subsystem, reflecting the obvious fact that the yaw rate and lateral acceleration cannot be independently controlled from a single steering angle. There exists no state feedback that keeps the outputs identically zero, since the system has no zero dynamics subsystem $x_d$.

If we add rear-axle steering by augmenting the $\hat{B}$ matrix with a column $[0, 0, 0, \frac{1}{v_i}]^T$, the Maple procedure returns the
SCB system matrices

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-\frac{c_1+c_2}{mT_\alpha} & -\frac{1}{T_\alpha} & -\frac{v_x}{T_\alpha} & 0 \\
0 & 0 & 0 & 0 \\
\frac{c_1-c_2}{J_\theta T_\theta} & 0 & \frac{1}{T_\alpha} & -\frac{1}{T_\alpha}
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0 & 0 \\
1 & 0 \\
0 & 0 \\
0 & 1
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix},
\]

\[
D = \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}.
\]

with dimensions 1, 0, 0, 3. This means that the first state of the system belongs to the zero dynamics \(x_\alpha\), and the remaining three states belong to the \(x_d\) subsystem. The \(x_d\) subsystem consists of two integrator chains; one of dimension one, and one of dimension two. We conclude that the system is both right- and left-invertible, due to the lack of \(x_b\) and \(x_c\) subsystems. Because \(A_{ii} = 0\), we see that the system has a zero at the origin. Hence, the relationship between the inputs and the outputs is non-minimum phase.

V. EXAMPLE: DC MOTOR WITH FRICTION

According to [15], a DC motor process can be described by the equations \(\dot{x} = \Phi x + \sigma_0 z/\zeta(x)\), where \(\Phi\) is the control input. The original system is described by the matrices $A$, $B$, $C$, and $D$. Inserting numerical values and using the Linear Systems Toolkit [9] yields the SCB matrices

\[
A \approx \begin{bmatrix}
-433.3 & -592.7 & 0 \\
0 & 0 & 1 \\
-1.1 \cdot 10^5 & -1.5 \cdot 10^5 & 95.9
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix},
\]

\[
C = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix},
\]

\[
D = \begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}.
\]

This reveals that a source of the conditioning problem is powers of the small parameter \(\sigma_1\) appearing in the denominators, even though it does not appear in any denominators in (8). In particular, we see that \(\sigma_1\) acts as a small regular perturbation that results in singularly perturbed zero dynamics, which happens when a regular perturbation reduces a system’s relative degree [17]. Setting \(\sigma_1 = 0\) results in a dramatically different structure, with the SCB consisting of a single integrator chain of length three. Proceeding with the observer gain selection based on this system, we obtain good results without using high gains.

VI. CONCLUDING REMARKS

The preceding example shows that the symbolic form of the SCB can be used to reveal structural bifurcations in linear systems due to parameter changes. Systematic ways of using symbolic representations of the SCB for this purpose is a topic of future research. Future research will also investigate application of symbolic SCB representations to topics where the SCB has previously been applied, such as squaring down of non-square systems and asymptotic time-scale assignment.

It is possible to transform the \(x_d\) subsystem so that the influence of \(x_a\) is matched with the input \(u_c\). Future versions will perform the extra step necessary to achieve this.

REFERENCES


