Structural Decomposition of Linear Multivariable Systems Using Symbolic Computations

Håvard Fjær Grip*
Department of Engineering Cybernetics
Norwegian University of Science and Technology
O.S. Bragstads plass 2D, 7491 Trondheim, Norway
Phone: +1 (509) 715-9195
E-mail: grip@itk.ntnu.no

Ali Saberi
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752, USA
Phone: +1 (509) 432-1138
E-mail: saberi@eecs.wsu.edu

Abstract

We introduce a procedure written in the mathematics software suite Maple, which transforms linear time-invariant 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. Throughout the procedure, transformations are constructed with the goal of avoiding unnecessary changes to the original states. The procedure is intended to complement numerical software algorithms developed by others for the same purpose. We discuss various system-theoretic aspects of the special coordinate basis as well as numerical issues related to the decomposition procedure, and illustrate use of the procedure by examples.

1 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, time scale assignment, disturbance rejection, H_2 control, and H_∞ control. It has also been used as a fundamental tool in the study of linear systems theory. For details on these topics, we refer to the books [2]–[6], all of which are based on the SCB, and references therein. Other topics include decoupling theory [1], factorization of linear systems [7], squaring down of non-square systems [1, 8], and model reduction [9].

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 [10]. 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 [11, 5]) and implemented as part of the *Linear Systems Toolkit* for *Matlab* [12]. 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

*Corresponding author

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 a modification to achieve a later version of the SCB that includes an additional structural property (see, e.g., [13]), and an extension to SCB for non-strictly proper systems [8]. The purpose of this paper is to introduce this procedure, and to explain how it is implemented using Maple and the *LinearAlgebra* package. The paper is also intended to serve as an introduction to the SCB, in particular for readers that might benefit from the possibility of working with symbolically represented systems in SCB form.

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. The work presented in this paper also constitutes the first step in a wider effort to apply 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.

1.1 Notation

We denote by $\operatorname{col}(\boldsymbol{z_1}, \ldots, \boldsymbol{z_n})$ the column vector obtained by stacking the column vectors $\boldsymbol{z_1}, \ldots, \boldsymbol{z_n}$. We denote by $\operatorname{diag}(M_1, \ldots, M_n)$ the matrix with submatrices $M_1, \ldots, M_n$ (not necessarily of the same dimensions) along the diagonal. We denote by I_n the $n \times n$ identity matrix. The symbol 0 may refer to the scalar number zero, or a zero matrix of appropriate dimensions.

2 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 [14]. 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

$$\dot{\hat{\boldsymbol{x}}} = \hat{A}\hat{\boldsymbol{x}} + \hat{B}\hat{\boldsymbol{u}}, \tag{1a}$$

$$\hat{\boldsymbol{y}} = \hat{C}\hat{\boldsymbol{x}} + \hat{D}\hat{\boldsymbol{u}}. \tag{1b}$$

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

For simplicity in the non-strictly proper case (i.e., $\hat{D} \neq 0$), we assume in this section that the input and output are partitioned as

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u_0} \\ \hat{\boldsymbol{u}}_1 \end{bmatrix} \text{ and } \hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y_0} \\ \hat{\boldsymbol{y}}_1 \end{bmatrix},$$

where $\boldsymbol{u_0}$ and $\boldsymbol{y_0}$ are of dimension m_0, and furthermore that $\hat{D}$ has the form $\hat{D} = \operatorname{diag}(I_{m_0}, 0)$. Then we may write

$$\hat{y} = \begin{bmatrix} \boldsymbol{y_0} \\ \hat{\boldsymbol{y}}_1 \end{bmatrix} = \begin{bmatrix} \hat{C}_0\hat{\boldsymbol{x}} + \boldsymbol{u_0} \\ \hat{C}_1\hat{\boldsymbol{x}} \end{bmatrix}, \tag{2}$$

where $\hat{C}_0$ 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: the output $\boldsymbol{y_0}$ is directly affected by $\boldsymbol{u_0}$, and the remainder of the output $\hat{\boldsymbol{y}}_1$ is not directly affected by any input. Note that by substituting $\boldsymbol{u_0} = \boldsymbol{y_0} - \hat{C}_0\hat{\boldsymbol{x}}$, we can write the system (1) in the alternative form

$$\dot{\hat{\boldsymbol{x}}} = (\hat{A} - \hat{B}_0\hat{C}_0)\hat{\boldsymbol{x}} + \hat{B}\begin{bmatrix} \boldsymbol{y_0} \\ \hat{\boldsymbol{u}}_1 \end{bmatrix}, \tag{3a}$$

$$\hat{\boldsymbol{y}} = \hat{C}\hat{\boldsymbol{x}} + \hat{D}\hat{\boldsymbol{u}}. \tag{3b}$$

where $\hat{B}_0$ consists of the left m_0 columns of $\hat{B}$. In the strictly proper case, $\hat{B}_0$ and $\hat{C}_0$ are nonexistent.

By nonsingular transformation of the state, output, and input, the system (1) can be transformed to the SCB. We use the symbols $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{u}$ 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 Γ_1, Γ_2, and Γ_3, and we write $\hat{\boldsymbol{x}} = \Gamma_1\boldsymbol{x}$, $\hat{\boldsymbol{y}} = \Gamma_2\boldsymbol{y}$, and $\hat{\boldsymbol{u}} = \Gamma_3\boldsymbol{u}$.

The state $\boldsymbol{x}$ is partitioned as $\boldsymbol{x} = \mathrm{col}(\boldsymbol{x_a}, \boldsymbol{x_b}, \boldsymbol{x_c}, \boldsymbol{x_d})$, where each component represents a particular subsystem described in the next section. The output is partitioned as $\boldsymbol{y} = \mathrm{col}(\boldsymbol{y_0}, \boldsymbol{y_d}, \boldsymbol{y_b})$, where $\boldsymbol{y_0}$ is the original output $\boldsymbol{y_0}$ from (1), $\boldsymbol{y_d}$ is the output from the $\boldsymbol{x_d}$ subsystem, and $\boldsymbol{y_b}$ is the output from the $\boldsymbol{x_b}$ subsystem. The input is partitioned as $\boldsymbol{u} = \mathrm{col}(\boldsymbol{u_0}, \boldsymbol{u_d}, \boldsymbol{u_c})$, where $\boldsymbol{u_0}$ is the original input $\boldsymbol{u_0}$ from (1), $\boldsymbol{u_d}$ is the input to the $\boldsymbol{x_d}$ subsystem, and $\boldsymbol{u_c}$ is the input to the $\boldsymbol{x_c}$ subsystem. The transformation Γ_3 is on the form $\mathrm{diag}(I_{m_0}, \bar{\Gamma}_3)$, where $\bar{\Gamma}_3$ is nonsingular.

2.1 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 $\boldsymbol{x_a}$ 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 $\boldsymbol{y_b}$ and $\boldsymbol{y_d}$ from the $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ subsystems.
- The $\boldsymbol{x_b}$ subsystem has a direct effect on the output $\boldsymbol{y_b}$, but it is not directly affected by any inputs. It may be affected, however, by the output $\boldsymbol{y_d}$ from the $\boldsymbol{x_d}$ subsystem. The $\boldsymbol{x_b}$ subsystem is observable from $\boldsymbol{y_b}$.
- The $\boldsymbol{x_c}$ subsystem is directly affected by the input $\boldsymbol{u_c}$, but it does not have a direct effect on any outputs. It may also be affected by the outputs $\boldsymbol{y_b}$ and $\boldsymbol{y_d}$ from the $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ subsystems, as well as the state $\boldsymbol{x_a}$. However, the influence from $\boldsymbol{x_a}$ is matched with the input $\boldsymbol{u_c}$. The $\boldsymbol{x_c}$ subsystem is controllable from $\boldsymbol{u_c}$.
- The $\boldsymbol{x_d}$ subsystem represents the infinite zero structure. This part of the system is directly affected by the input $\boldsymbol{u_d}$, and it also affects the output $\boldsymbol{y_d}$ directly. The $\boldsymbol{x_d}$ subsystem can be further partitioned into m_d single-input single-output (SISO) subsystems $\boldsymbol{x_i}$ for $i = 1, \dots, m_d$. Each of these subsystems consists of a chain of integrators of length q_i, from the i'th element of $\boldsymbol{u_d}$ to the i'th element of $\boldsymbol{y_d}$. Each integrator chain may be affected at each stage by the output $\boldsymbol{y_d}$ from the $\boldsymbol{x_d}$ 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 $\boldsymbol{x_d}$ subsystem is observable from $\boldsymbol{y_d}$, and controllable from $\boldsymbol{u_d}$.

The structure of strictly proper SCB systems is summarized in Table 1. For non-strictly proper systems the structure is the same, except for the existence of the direct-feedthrough output $\boldsymbol{y_0}$, which is directly affected by the input $\boldsymbol{u_0}$, and can be affected by any of the states of the system. It can also affect all the states of the system.

2.2 SCB Equations

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

$$\dot{\boldsymbol{x}}_{\boldsymbol{a}} = A_{aa}\boldsymbol{x_a} + B_{a0}\boldsymbol{y_0} + L_{ad}\boldsymbol{y_d} + L_{ab}\boldsymbol{y_b}, \tag{4a}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{b}} = A_{bb}\boldsymbol{x_b} + B_{b0}\boldsymbol{y_0} + L_{bd}\boldsymbol{y_d}, \tag{4b}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{c}} = A_{cc}\boldsymbol{x_c} + B_{c0}\boldsymbol{y_0} + L_{cd}\boldsymbol{y_d} + L_{cb}\boldsymbol{y_b} + B_c(\boldsymbol{u_c} + E_{ca}\boldsymbol{x_a}), \tag{4c}$$

$$\dot{\boldsymbol{x}}_{\boldsymbol{i}} = A_{q_i}\boldsymbol{x_i} + B_{d0}\boldsymbol{y_0} + L_{id}\boldsymbol{y_d} + B_{q_i}(u_i + E_{ia}\boldsymbol{x_a} + E_{ib}\boldsymbol{x_b} + E_{ic}\boldsymbol{x_c} + E_{id}\boldsymbol{x_d}), \tag{4d}$$

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

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

$$y_i = C_{q_i}\boldsymbol{x_i}, \quad i = 1, \ldots, m_d, \tag{5b}$$

$$\boldsymbol{y_b} = C_b\boldsymbol{x_b}. \tag{5c}$$

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

$$A_{q_i} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B_{q_i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad C_{q_i} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}.$$

The pair (C_b, A_{bb}) is observable, and the pair (A_{cc}, B_c) is controllable. In the strictly proper case, the input $\boldsymbol{u_0}$ and output $\boldsymbol{y_0}$ are nonexistent, as are the matrices B_{a0}, B_{b0}, B_{c0}, B_{d0}, C_{0a}, C_{0b}, C_{0c}, and C_{0d}.

2.3 Compact Form

We may write (4) as

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\begin{bmatrix} \boldsymbol{y_0} \\ \boldsymbol{u_d} \\ \boldsymbol{u_c} \end{bmatrix}, \tag{6a}$$

$$\boldsymbol{y} = C\boldsymbol{x} + D\boldsymbol{u}, \tag{6b}$$

Subsystem	**Input**	**Output**	**Interconnections**	**Remarks**
$\boldsymbol{x_a}$	—	—	$\boldsymbol{y_b}$, $\boldsymbol{y_d}$	Zero dynamics
$\boldsymbol{x_b}$	—	$\boldsymbol{y_b}$	$\boldsymbol{y_d}$	Observable
$\boldsymbol{x_c}$	$\boldsymbol{u_c}$	—	$\boldsymbol{y_b}$, $\boldsymbol{y_d}$, $\boldsymbol{x_a}^\star$	Controllable
$\boldsymbol{x_d}$	$\boldsymbol{u_d}$	$\boldsymbol{y_d}$	$\boldsymbol{x_a}^\star$, $\boldsymbol{x_b}^\star$, $\boldsymbol{x_c}^\star$	Observable and controllable

*Matched with input

Table 1: Summary of strictly proper SCB structure. The *Interconnections* column indicates influences from other subsystems.

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

$$A = \begin{bmatrix} A_{aa} & L_{ab}C_b & 0 & L_{ad}C_d \\ 0 & A_{bb} & 0 & L_{bd}C_d \\ B_c E_{ca} & L_{cb}C_b & A_{cc} & L_{cd}C_d \\ B_d E_{da} & B_d E_{db} & B_d E_{dc} & A_{dd} \end{bmatrix}, \qquad B = \begin{bmatrix} B_{a0} & 0 & 0 \\ B_{b0} & 0 & 0 \\ B_{c0} & 0 & B_c \\ B_{d0} & B_d & 0 \end{bmatrix},$$
$$C = \begin{bmatrix} C_{0a} & C_{0b} & C_{0c} & C_{0d} \\ 0 & 0 & 0 & C_d \\ 0 & C_b & 0 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} I_{m_0} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $A_{dd} = \operatorname{diag}(A_{q_1}, \ldots, A_{q_{m_d}}) + L_{dd}C_d + B_d E_{dd}$, $B_d = \operatorname{diag}(B_{q_1}, \ldots, B_{q_{m_d}})$, $C_d = \operatorname{diag}(C_{q_1}, \ldots, C_{q_{m_d}})$, $L_{dd} = [L_{1d}^\mathsf{T}, \ldots, L_{m_d d}^\mathsf{T}]^\mathsf{T}$, $E_{da} = [E_{1a}^\mathsf{T}, \ldots, E_{m_d a}^\mathsf{T}]^\mathsf{T}$, 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 $\hat{D}$ and the SCB matrices A, B, C, and D from (6), substitute $\hat{\boldsymbol{x}} = \Gamma_1 \boldsymbol{x}$, $\hat{\boldsymbol{y}} = \Gamma_2 \boldsymbol{y}$, and $\hat{\boldsymbol{u}} = \Gamma_3 \boldsymbol{u}$ in equation (3). Also, note that since Γ_3 is of the form $\operatorname{diag}(I_{m_0}, \bar{\Gamma}_3)$, we can make the substitution $\operatorname{col}(\boldsymbol{y_0}, \hat{\boldsymbol{u}}_\mathbf{1}) = \Gamma_3 \operatorname{col}(\boldsymbol{y_0}, \boldsymbol{u_d}, \boldsymbol{u_c})$. We then obtain the equations

$$\begin{aligned} \dot{\boldsymbol{x}} &= \Gamma_1^{-1}(\hat{A} - \hat{B}_0\hat{C}_0)\Gamma_1 \boldsymbol{x} + \Gamma_1^{-1}\hat{B}\Gamma_3 \begin{bmatrix} \boldsymbol{y_0} \\ \boldsymbol{u_d} \\ \boldsymbol{u_c} \end{bmatrix}, \\ \boldsymbol{y} &= \Gamma_2^{-1}\hat{C}\Gamma_1 \boldsymbol{x} + \Gamma_2^{-1}\hat{D}\Gamma_3 \boldsymbol{u}. \end{aligned}$$

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

2.4 Pre-Transformation of Non-Strictly Proper Systems

We assumed initially that the input and output vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{y}}$ 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 $\tilde{\boldsymbol{u}}$, output $\tilde{\boldsymbol{y}}$, input matrix $\tilde{B}$, and output matrices $\tilde{C}$ and $\tilde{D}$. Then there are nonsingular transformations U and Y such that $\tilde{\boldsymbol{u}} = U\hat{\boldsymbol{u}}$ and $\tilde{\boldsymbol{y}} = Y\hat{\boldsymbol{y}}$, where $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{y}}$ have the structure required in (2). The dimension m_0 of $\boldsymbol{u_0}$ and $\boldsymbol{y_0}$ is the rank of $\tilde{D}$. The matrices $\hat{B}$, $\hat{C}$, and $\hat{D}$ are obtained from $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$ by $\hat{B} = \tilde{B}U$, $\hat{C} = Y^{-1}\tilde{C}$, and $\hat{D} = Y^{-1}\tilde{D}U$. Our Maple procedure, in addition to returning the matrices A, B, C, and D of the SCB system, the transformations Γ_1, Γ_2, and Γ_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 in (1), (2).

3 Properties of the SCB

The SCB is closely related to the canonical form of Morse [15], which is obtained through transformations of the state, input, and output spaces, and the application of state feedback and output injection. A system in the canonical form of Morse consists of four decoupled subsystems that reflect essential geometric properties of the original system. The SCB form of a system largely reflects the same properties; however, the SCB is obtained through transformations of the state, input, and output spaces alone, without the application of state feedback and output injection. Thus, the SCB is merely a representation of the original system in a different coordinate basis, and it can therefore be used directly for design purposes.

Some properties of the SCB, which correspond directly to properties of the canonical form of Morse, are the following:

- The invariant zeros of the system (1) are the eigenvalues of the matrix A_{aa}. Hence, the system is minimum-phase if, and only if, the eigenvalues of A_{aa} are located in the open left-half complex plane.
- The system (1) is right-invertible if, and only if, the subsystem $\boldsymbol{x_b}$ is non-existent.
- The system (1) is left-invertible if, and only if, the subsystem $\boldsymbol{x_c}$ is non-existent.
- The system (1) is invertible if, and only if, both the subsystem $\boldsymbol{x_b}$ and the subsystem $\boldsymbol{x_c}$ are nonexistent.
- The system (1) has m_0 infinite zeros of order 0 and $i\bar{q}_i$ infinite zeros of order i, where $\bar{q}_i$ is the of number integrator chains of length i in the $\boldsymbol{x_d}$ subsystem.

By studying the dynamics of the $\boldsymbol{x_a}$ subsystem and its connections to the rest of the system, one obtains a precise description of the invariant zero dynamics of the system and the classes of input signals that may be blocked by these zeros. The information thus obtained goes beyond what can be obtained through the notions of state and input *pseudo zero directions* (see [16, 13]).

The representation of the infinite zero structure through integrator chains in the $\boldsymbol{x_d}$ subsystem allows for the explicit construction of high-gain controllers and observers in a general multiple-input multiple-output setting (see, e.g., [17]). This removes unnecessary restrictions of square-invertibility and uniform relative degree that are found in much of the high-gain literature.

3.1 Connection to Geometry Theory

Geometry theory is concerned with the study of subspaces of the state space with certain invariance properties, for example, A-invariant subspaces (which remain invariant under the unforced motion of the system), (A, B)-invariant subspaces (which can be made invariant by the proper application of state feedback), and (C, A) invariant subspaces (which can be made invariant by the proper application of output injection) (see, e.g., [18, 19]). Prominent examples of A-invariant subspaces are the controllable subspace (i.e., the image of the controllability matrix) and the unobservable subspace (the kernel of the observability matrix).

The development of geometry theory has in large part been motivated by the challenge of decoupling disturbance inputs from the outputs of a system, either exactly or approximately. Toward this end, a number of subspaces have been identified, which can be related to the partitioning in the SCB. Of particular importance in the context of control design for exact disturbance decoupling are the *weakly unobservable subspace*, which, by the proper selection of state feedback, can be made not to affect the outputs; and the *controllable weakly unobservable subspace*, which has the additional property that the dynamics restricted to this subspace is controllable. Of particular importance in the context of observer design for exact disturbance decoupling are the *strongly controllable subspace*, which, by the proper selection of output injection, is such that its quotient space can be rendered unaffected by the system inputs; and the *distributionally weakly unobservable subspace*, which has the additional property that the dynamics restricted to its quotient space is observable.

We denote by $\mathcal{X}_a$, $\mathcal{X}_b$, $\mathcal{X}_c$, and $\mathcal{X}_d$ the subspaces spanned by the states $\boldsymbol{x_a}$, $\boldsymbol{x_b}$, $\boldsymbol{x_c}$, and $\boldsymbol{x_d}$, and by $\oplus$ the direct sum of two subspaces that intersect only at the origin. The subspaces mentioned above can then be related to the SCB as follows:

- The weakly unobservable subspace is given by $\mathcal{X}_a \oplus \mathcal{X}_c$.

- The controllable weakly unobservable subspace is given by $\mathcal{X}_c$.
- The strongly controllable subspace is given by $\mathcal{X}_c \oplus \mathcal{X}_d$.
- The distributionally weakly unobservable subspace is given by $\mathcal{X}_a \oplus \mathcal{X}_c \oplus \mathcal{X}_d$.

A list of further subspaces identified in geometry theory and their relationship to the SCB can be found in [13].

The SCB provides a more direct and tangible path to disturbance decoupling design than the somewhat abstract notions of geometry theory. For example, geometry theory tells us that a disturbance entering into the weakly unobservable subspace can be decoupled from the outputs by the proper selection of state feedback. In the SCB the weakly unobservable subspace is represented by the state variables $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$; thus, a disturbance affecting only $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$ can be decoupled from the outputs. This decoupling is achieved by selecting the state feedback $\boldsymbol{u_0} = -C_{0a}\boldsymbol{x_a} - C_{0c}\boldsymbol{x_c} + \boldsymbol{v_0}$, $u_i = -E_{ia}\boldsymbol{x_a} - E_{ic}\boldsymbol{x_c} + v_i$, $i = 1, \dots, m_d$, where $\boldsymbol{v_0}$ and v_i, as well as $\boldsymbol{u_c}$, can be chosen freely. It can be verified by direction inspection of (4) that this state feedback cancels the influence of $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$ on the rest of the system, and therefore on the outputs. With the help of symbolic transformations, such decoupling design can be carried out directly on systems with a symbolic representation.

3.2 Further Properties

Some useful connections can be made between the SCB representation of a system and the properties of controllability, stabilizability, observability, and detectability:

- The system (1) is controllable (stabilizable) if, and only if, the pair

$$\left(\begin{bmatrix} A_{aa} & L_{ab}C_b \\ 0 & A_{bb} \end{bmatrix}, \begin{bmatrix} B_{a0} & L_{ad} \\ B_{b0} & L_{bd} \end{bmatrix} \right)$$

  is controllable (stabilizable).

- The system (1) is observable (detectable) if, and only if, the pair

$$\left(\begin{bmatrix} C_{0a} & C_{0c} \\ E_{da} & E_{dc} \end{bmatrix}, \begin{bmatrix} A_{aa} & 0 \\ B_c E_{ca} & A_{cc} \end{bmatrix} \right)$$

  is observable (detectable).

- The system (1) is stabilizable if it is right-invertible and minimum-phase (i.e., the $\boldsymbol{x_b}$ subsystem is nonexistent and the eigenvalues of A_{aa} are in the open left-half plane).
- The system (1) is detectable if it is left-invertible and minimum-phase (i.e., the $\boldsymbol{x_c}$ subsystem is nonexistent and the eigenvalues of A_{aa} are in the open left-half plane).

The subsystem partitioning of the SCB remains the same when state feedback and output injection is applied to the system. This is in contrast to the system obtained by a Kalman decomposition, which is partitioned according to the properties of controllability and observability.

4 Maple Procedure

Our Maple procedure is invoked as follows:

```
A, B, C, D, G1, G2, G3, U, Y, dim := scb(Ai, Bi, Ci, Di);
```

The inputs Ai, Bi, Ci, and Di are system matrices describing a general multivariable LTI system. The outputs A, B, C, and D are the system matrices describing a corresponding SCB system. The outputs G1, G2, and G3 are the transformation matrices Γ_1, Γ_2, and Γ_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), (2), as described in Section 2.4. Finally, the output dim is a list of four integers representing the dimensions of the $\boldsymbol{x_a}$, $\boldsymbol{x_b}$, $\boldsymbol{x_c}$, and $\boldsymbol{x_d}$ subsystems, in that order. The Maple source code is available from [20].

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 $\boldsymbol{y_{i0}}$, given by the expression $\boldsymbol{y_{i0}} = C_i\hat{\boldsymbol{x}}$, is represented internally by a **Matrix** data structure containing C_i. 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 **convert** function. In many cases, we need to store a whole list of matrices, representing variables obtained during successive iterations of a particular part of the algorithm. To do this, we use the Maple data structures **Vector** and **Matrix**, which can be used to store vectors or matrices whose elements are **Matrix** data structures.

4.1 Strictly Proper Case

The algorithm for strictly proper systems is implemented as **scbSP**. The first part of this algorithm identifies the two subsystems that directly influence the outputs, namely the $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ 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 **scbSP** procedure be represented by the state equations $\dot{\hat{\boldsymbol{x}}} = \hat{A}\hat{\boldsymbol{x}} + \hat{B}\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{y}} = \hat{C}\hat{\boldsymbol{x}}$. In the first iteration we start with the output $\boldsymbol{y_{10}} = \hat{C}\hat{\boldsymbol{x}}$, and determine whether its derivative $\dot{\boldsymbol{y}}_{\boldsymbol{10}} = \hat{C}\hat{A}\hat{\boldsymbol{x}} + \hat{C}\hat{B}\hat{\boldsymbol{u}}$ depends on any part of the input $\hat{\boldsymbol{u}}$. If so, we use a transformation $\bar{S}_1$ 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 $\boldsymbol{x_d}$ subsystem. A transformed part of the output derivative that is not directly influenced by the input is denoted $\tilde{C}_1\hat{\boldsymbol{x}}$, and is processed further. We use a transformation $\bar{\phi}_1$ to separate out any part of $\tilde{C}_1\hat{\boldsymbol{x}}$ that is linearly dependent on $\boldsymbol{y_{10}}$. This will create states that are part of the $\boldsymbol{x_b}$ subsystem. After the linearly dependent components are separated out, the remaining part of the output derivative is given the name $\boldsymbol{y_{20}}$. In the next iteration we process $\boldsymbol{y_{20}}$ in the same fashion as $\boldsymbol{y_{10}}$, to identify integrator chains of length two, and possibly further additions to the $\boldsymbol{x_b}$ subsystem. The algorithm continues in this fashion until the outputs are exhausted.

4.1.1 Constructing Transformation Matrices

When implementing these steps in Maple, the main part of each iteration consists of constructing transformation matrices $\bar{S}_i$ and $\bar{\phi}_i$. 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 $\bar{D}_{i-1}$ of dimension $\bar{q}_{i-1} \times m$ of maximal rank $\bar{q}_{i-1}$, let $\bar{q}_i$

be the rank of $[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}}$, and let $\mathfrak{q}_i = \bar{\mathfrak{q}}_i - \bar{\mathfrak{q}}_{i-1}$. Find a nonsingular matrix $\bar{S}_i$ such that

$$\bar{S}_i \begin{bmatrix} \bar{D}_{i-1} \\ C_i\hat{B} \end{bmatrix} = \begin{bmatrix} \bar{D}_{i-1} \\ \hat{D}_i \\ 0 \end{bmatrix}, \quad \bar{S}_i = \begin{bmatrix} I_{\bar{\mathfrak{q}}_i-1} & 0 \\ S_{ia} & S_i \end{bmatrix}, \quad S_{ia} = \begin{bmatrix} 0 \\ S_{ib} \end{bmatrix}, \quad S_i = \begin{bmatrix} S_{i1} \\ S_{i2} \end{bmatrix},$$

where $\hat{D}_i$ is a $\mathfrak{q}_i \times m$ matrix of maximal rank, and where S_{i1}, S_{i2}, and S_{ib} are of dimensions $\mathfrak{q}_i \times p_i$, $(p_i - \mathfrak{q}_i) \times p_i$, and $(p_i - \mathfrak{q}_i) \times \bar{\mathfrak{q}}_{i-1}$. The meaning of the various dimensions is not important in this context. In general, $\bar{S}_i$ is not unique.

The rank of the matrix $[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}}$ can be obtained with the **Rank** function in the LinearAlgebra package. To construct the matrix $\bar{S}_i$, the first observation we make is that, since $S_{ib}\bar{D}_{i-1} + S_{i2}C_i\hat{B} = 0$, the rows of the matrix $[S_{ib}, S_{i2}]$ must belong to the left null space of $[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}}$. If $[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}}$ has full rank $\bar{\mathfrak{q}}_{i-1} + p_i$, then S_{ib} and S_{i2} are empty matrices, and we may select $S_{ia} = 0$ and $S_{i1} = I_{p_i}$. Otherwise, we can obtain a set of linearly independent basis vectors for the left null space of $[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}}$, or equivalently, for the right null space of its transpose, using the **NullSpace** function of the LinearAlgebra package. The transpose of the basis vectors can then be stacked to form the matrix $[S_{ib}, S_{i2}]$, which can be split up to form S_{ib} and S_{i2}. 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 **ReducedRowEchelonForm** function 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_{i2} has maximal rank, which is confirmed as follows: if any of the rows of S_{i2} are linearly dependent, a linear combination of rows in $[S_{ib}, S_{i2}]$ can be constructed to create a row vector $\boldsymbol{v}$ such that $\boldsymbol{v}[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}} = 0$, where the rightmost p_i columns of $\boldsymbol{v}$ are zero. However, since the rows of $\bar{D}_{i-1}$ are linearly independent, this implies that $\boldsymbol{v} = 0$, which in turn implies that $[S_{ib}, S_{i2}]$ must have linearly dependent rows. Since this is not the case, S_{i2} must have maximal rank.

We continue by constructing the matrix S_{i1}. Nonsingularity of S_i requires that the rows of S_{i1} must be linearly independent of the rows of S_{i2}. One way to produce S_{i1} is to choose its rows to be orthogonal to the rows of S_{i2}, which can be achieved by using a basis for the right null space of S_{i2}. However, since the matrix $\bar{S}_i$ 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 avoids unnecessary changes to the original states, and thus it generally produces more appealing solutions. We therefore construct S_{i1} by the following procedure: we start by initializing S_{i1} as the identity matrix of dimension $p_i \times p_i$. We then create a reduced-row echelon form of S_{i2}, 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_{i1} corresponding to the column with the leading 1. This ensures that $S_i = [S_{i1}^{\mathsf{T}}, S_{i2}^{\mathsf{T}}]^{\mathsf{T}}$ is nonsingular, with S_{i1} consisting of zeros except for a single element equal to 1 on each row. The construction of $\bar{S}_i$ is now easily completed.

At each step, we must also construct a nonsingular matrix $\bar{\phi}_i$. The problem of finding this matrix is analogous to the problem of finding $\bar{S}_i$, and we therefore use the same procedure. Finding the transformations $\bar{S}_i$ and $\bar{\phi}_i$ constitute the most important part of finding the states $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$. After $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ are identified, finding the output transformation Γ_2 is straightforward, based on [1]. We also find an input transformation Γ_3' based on [1] and write $\hat{u} = \Gamma_3'[\boldsymbol{u_d}^{\mathsf{T}}, \boldsymbol{u_c'}^{\mathsf{T}}]^{\mathsf{T}}$, where $\boldsymbol{u_c'}$ is a temporary input. Unlike [1], we shall apply a further transformation to $\boldsymbol{u_c'}$ to achieve an input $\boldsymbol{u_c}$ that is matched with the influence from $\boldsymbol{x_a}$ on the right-hand side of the $\boldsymbol{x_c}$ equation.

4.2 Constructing the $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$ States

After finding the transformations from the original states to the $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ states, the next step is to find a transformation to a temporary state vector $\boldsymbol{x_s}$ that will be further decomposed into the states $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$. The requirements on $\boldsymbol{x_s}$ is that it must be linearly independent of the already identified states $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$, so that $\boldsymbol{x_s}$, $\boldsymbol{x_b}$, and $\boldsymbol{x_d}$ together span the entire state space; and that its derivative $\dot{\boldsymbol{x}}_{\boldsymbol{s}}$ must only depend on $\boldsymbol{x_s}$ itself, plus $\boldsymbol{y_b}$, $\boldsymbol{y_d}$, and $\boldsymbol{u_c'}$, because those are the only quantities allowed in the derivatives of $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$ in the strictly proper case.

Suppose that $\mathrm{col}(\boldsymbol{x_b}, \boldsymbol{x_d}) = \Gamma_{bd}\hat{\boldsymbol{x}}$. The procedure for finding $\boldsymbol{x_s}$ is to start with a temporary state vector $\boldsymbol{x_s^0} = \Gamma_s^0 \hat{\boldsymbol{x}}$ that is linearly independent of $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$. Hence, we select Γ_s^0 such that $[\Gamma_s^{0\mathsf{T}}, \Gamma_{bd}^{\mathsf{T}}]^{\mathsf{T}}$ is nonsingular. To do so in our Maple procedure, we use the same technique as for finding S_{i1} based on S_{i2} in Section 4.1.1.

The derivative of $\boldsymbol{x_s^0}$, written in terms of the states $\boldsymbol{x_s^0}$, $\boldsymbol{x_b}$, and $\boldsymbol{x_d}$, and the inputs $\boldsymbol{u_c'}$ and $\boldsymbol{u_d}$, can be written as

$$\dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\boldsymbol{0}} = A^0 \begin{bmatrix} \boldsymbol{x_s^0} \\ \boldsymbol{x_b} \\ \boldsymbol{x_d} \end{bmatrix} + B^0 \begin{bmatrix} \boldsymbol{u_d} \\ \boldsymbol{u_c'} \end{bmatrix} = A_s^0 \boldsymbol{x_s^0} + A_b^0 \boldsymbol{x_b} + A_d^0 \boldsymbol{x_d} + B_d^0 \boldsymbol{u_d} + B_c^0 \boldsymbol{u_c'},$$

for some matrices $A^0 = [A_s^0, A_b^0, A_d^0]$ and $B^0 = [B_d^0, B_c^0]$. In our Maple procedure, we can easily calculate $A^0 = \Gamma_s^0 \hat{A}([\Gamma_s^{0\mathsf{T}}, \Gamma_{bd}^{\mathsf{T}}]^{\mathsf{T}})^{-1}$ and $B^0 = \Gamma_s^0 \hat{B} \Gamma_3'$, and then extract the matrices A_s^0, A_b^0, A_c^0, B_d^0, and B_c^0. To do so, we use the **MatrixInverse** function of the LinearAlgebra package.

To conform with the SCB, we need to modify $\boldsymbol{x_s^0}$ to eliminate the input $\boldsymbol{u_d}$ in $\dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\boldsymbol{0}}$. To eliminate $\boldsymbol{u_d}$, we create a temporary state vector $\boldsymbol{x_{d0}} = \Gamma_{d0}\hat{\boldsymbol{x}}$, consisting of the lowermost level of each integrator chain in the $\boldsymbol{x_d}$ subsystem (that is, the point where the input enters the integrator chain). According to (4), we then have $\dot{\boldsymbol{x}}_{\boldsymbol{d0}} = \boldsymbol{u_d} + A_{d0}[\boldsymbol{x_s^0}^{\mathsf{T}}, \boldsymbol{x_b}^{\mathsf{T}}, \boldsymbol{x_d}^{\mathsf{T}}]^{\mathsf{T}}$, for some matrix A_{d0}. Therefore, by defining a new temporary state $\boldsymbol{x_s^1} = \boldsymbol{x_s^0} - B_d^0 \boldsymbol{x_{d0}}$, we have $\dot{\boldsymbol{x}}_{\boldsymbol{s}}^{\boldsymbol{1}} = (A^0 - B_d^0 A_{d0})[\boldsymbol{x_s^0}^{\mathsf{T}}, \boldsymbol{x_b}^{\mathsf{T}}, \boldsymbol{x_d}^{\mathsf{T}}]^{\mathsf{T}} + B_c^0 \boldsymbol{u_c'}$. Hence, the derivative of the new temporary state vector $\boldsymbol{x_s^1}$ is independent of $\boldsymbol{u_d}$, bringing us one step closer to obtaining $\boldsymbol{x_s}$. The elimination procedure is continued in a similar fashion, as described in [1], until we obtain a state $\boldsymbol{x_s}$ such that $\dot{\boldsymbol{x}}_{\boldsymbol{s}}$ depends only on $\boldsymbol{x_s}$, $\boldsymbol{y_b}$, $\boldsymbol{y_d}$, and $\boldsymbol{u_c'}$.

The final step is to decompose $\boldsymbol{x_s}$ into two subsystems, $\boldsymbol{x_a}$ and $\boldsymbol{x_c}$, and to transform the input $\boldsymbol{u_c'}$ into $\boldsymbol{u_c}$, in such a way that $\boldsymbol{x_a}$ is unaffected by $\boldsymbol{u_c}$ and $\boldsymbol{x_c}$ is controllable from $\boldsymbol{u_c}$. Furthermore, the influence of $\boldsymbol{x_a}$ on $\boldsymbol{x_c}$ should be matched with $\boldsymbol{u_c}$, as seen in (4). If $\boldsymbol{u_c'}$ is nonexistent, then we simply set $\boldsymbol{x_a} = \boldsymbol{x_s}$. If $\boldsymbol{u_c'}$ does exist, we proceed by first finding the derivative $\dot{\boldsymbol{x}}_{\boldsymbol{s}} = A_{ss}\boldsymbol{x_s} + L_{sb}\boldsymbol{y_b} + L_{sd}\boldsymbol{y_d} + B_{sc}\boldsymbol{u_c'}$, for some matrices A_{ss}, L_{sb}, L_{sd}, and B_{sc}. We then obtain the proper transformations by calling **scbSP** recursively on the transposed system with system matrix A_{ss}^{T}, output matrix B_{sc}^{T}, and an empty input matrix. This recursive call returns a system consisting only of an $\boldsymbol{x_a}$ and an $\boldsymbol{x_b}$ subsystem. It is easily confirmed that, when transposed back again, this system has the desired structure. We therefore let $[\boldsymbol{x_a}^{\mathsf{T}}, \boldsymbol{x_c}^{\mathsf{T}}]^{\mathsf{T}} = \Gamma_1^{\star\mathsf{T}} \boldsymbol{x_s}$ and $\boldsymbol{u_c} = \Gamma_2^{\star\mathsf{T}} \boldsymbol{u_c'}$, where $\Gamma_1^\star$ and $\Gamma_2^\star$ are the state and output transformations returned by the recursive call.

4.3 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 2.4. Suppose that the matrices passed to the procedure **scb** are $\hat{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$. We need to find nonsingular U and Y such that, according to Section 2.4, $\hat{B} = \tilde{B}U$, $\hat{C} = Y^{-1}\tilde{C}$, and $\hat{D} = Y^{-1}\tilde{D}U$, where $\hat{D}$ is of the form $\mathrm{diag}(I_{m_0}, 0)$. The rank m_0 of $\tilde{D}$ is found using the **Rank** function. Let $Y^{-1} = [Y_1^{\mathsf{T}}, Y_2^{\mathsf{T}}]^{\mathsf{T}}$, where Y_1 has m_0 rows. Then we have the equations

$$Y^{-1}\tilde{D}U = \begin{bmatrix} Y_1 \tilde{D} U \\ Y_2 \tilde{D} U \end{bmatrix} = \begin{bmatrix} I_{m_0} & 0 \\ 0 & 0 \end{bmatrix}.$$

To solve these equations, we choose the rows of Y_2 from the left null space of $\tilde{D}$, using the functions **NullSpace** and **ReducedRowEchelonForm** as before; and we select Y_1 such that $[Y_1^\top, Y_2^\top]^\top$ is nonsingular, using the same procedure as for finding S_{i1} given S_{i2} in Section 4.1.1. This leaves us to solve the equation $Y_1 \tilde{D} U = [I_{m_0}, 0]$ with respect to some nonsingular U. Let $U^{-1} = [U_1^\top, U_2^\top]^\top$ such that U_1 has m_0 rows. We select $U_1 = Y_1 \tilde{D}$, and we select U_2 such that $[U_1^\top, U_2^\top]^\top$ is nonsingular, by the same procedure as before. It is then straightforward to confirm that $Y_1 \tilde{D} U = [I_{m_0}, 0]$. We can now calculate the matrices $\hat{B}$, $\hat{C}$, and $\hat{D}$ that conform with the required structure of (1), (2).

Let $\hat{B}_0$ consist of the left m_0 columns of $\hat{B}$, and let $\hat{B}_1$ consist of the remaining columns of $\hat{B}$. Similar to (3), we can write the system equations (1) as

$$\dot{\hat{\boldsymbol{x}}} = (\hat{A} - \hat{B}_0 \hat{C}_0)\hat{\boldsymbol{x}} + \hat{B}_0 \boldsymbol{y_0} + \hat{B}_1 \hat{\boldsymbol{u}}_{\boldsymbol{1}}, \tag{7a}$$

$$\boldsymbol{y_0} = \hat{C}_0 \hat{\boldsymbol{x}} + \boldsymbol{u_0}, \tag{7b}$$

$$\hat{\boldsymbol{y}}_{\boldsymbol{1}} = \hat{C}_1 \hat{\boldsymbol{x}}. \tag{7c}$$

Suppose we obtain the SCB form of the strictly proper system described by the matrices $(\hat{A} - \hat{B}_0 \hat{C}_0)$, $\hat{B}_1$, and $\hat{C}_1$, by invoking the procedure **scbSP**, and suppose the transformation matrices returned for this system are $\bar{\Gamma}_1$, $\bar{\Gamma}_2$, and $\bar{\Gamma}_3$. Substituting $\hat{\boldsymbol{x}} = \bar{\Gamma}_1 \boldsymbol{x}$, $\hat{\boldsymbol{y}}_{\boldsymbol{1}} = \bar{\Gamma}_2 [\boldsymbol{y_d}^\top, \boldsymbol{y_b}^\top]^\top$, and $\hat{\boldsymbol{u}}_{\boldsymbol{1}} = \bar{\Gamma}_3 [\boldsymbol{u_d}^\top, \boldsymbol{u_c}^\top]^\top$ in (7) yields

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \bar{\Gamma}_1^{-1} (\hat{A} - \hat{B}_0 \hat{C}_0) \bar{\Gamma}_1 \boldsymbol{x} + \bar{\Gamma}_1^{-1} \hat{B}_0 \boldsymbol{y_0} + \bar{\Gamma}_1^{-1} \hat{B}_1 \bar{\Gamma}_3 \begin{bmatrix} \boldsymbol{u_d} \\ \boldsymbol{u_c} \end{bmatrix}, \\
\boldsymbol{y_0} &= \hat{C}_0 \bar{\Gamma}_1 \boldsymbol{x} + \boldsymbol{u_0}, \\
\begin{bmatrix} \boldsymbol{y_d} \\ \boldsymbol{y_b} \end{bmatrix} &= \bar{\Gamma}_2^{-1} \hat{C}_1 \bar{\Gamma}_1 \boldsymbol{x}.
\end{aligned}$$

It is easily confirmed that this system conforms to the SCB, by defining $A = \bar{\Gamma}_1^{-1} (\hat{A} - \hat{B}_0 \hat{C}_0) \bar{\Gamma}_1$, $B = \bar{\Gamma}_1^{-1} [\hat{B}_0, \hat{B}_1 \bar{\Gamma}_3]$, $C = [\hat{C}_0^\top, (\bar{\Gamma}_2^{-1} \hat{C}_1)^\top]^\top \bar{\Gamma}_1$, and $D = \mathrm{diag}(I_{m_0}, 0)$. Defining the transformations for the non-strictly proper system as $\Gamma_1 = \bar{\Gamma}_1$, $\Gamma_2 = \mathrm{diag}(I_{m_0}, \bar{\Gamma}_2)$, and $\Gamma_3 = \mathrm{diag}(I_{m_0}, \bar{\Gamma}_3)$, we obtain $A = \Gamma_1^{-1} (\hat{A} - \hat{B}_0 \hat{C}_0) \Gamma_1$, $B = \Gamma_1^{-1} \hat{B} \Gamma_3$, $C = \Gamma_2^{-1} \hat{C} \Gamma_1$, and $D = \Gamma_2^{-1} \hat{D} \Gamma_3$, which are the proper expressions relating the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{D}$ to the SCB matrices (see Section 2.3).

5 Examples

In this section, we apply the decomposition procedure to several example systems.

5.1 Linear Single-Track Model

A widely used model for the lateral dynamics of a car is the linear single-track model (see, e.g., [21]). For a car on a horizontal surface, this model is described by the equations

$$\begin{aligned}
\dot{v}_y &= \frac{1}{m}(F_\mathrm{f} + F_\mathrm{r}) - r v_x, \\
\dot{r} &= \frac{1}{J}(l_\mathrm{f} F_\mathrm{f} - l_\mathrm{r} F_\mathrm{r}),
\end{aligned}$$

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 around the vertical axis through the car's center of gravity; l_f and l_r are the longitudinal distances from the center of gravity to the front and rear axles; and F_f 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

$$\dot{F}_{\mathrm{f}} = \frac{c_{\mathrm{f}}}{T_{\mathrm{r}}}\left(\delta_{\mathrm{f}} - \frac{v_y}{v_x} - l_{\mathrm{f}}\frac{r}{v_x}\right) - \frac{1}{T_{\mathrm{r}}}F_{\mathrm{f}},$$
$$\dot{F}_{\mathrm{r}} = \frac{c_{\mathrm{r}}}{T_{\mathrm{r}}}\left(-\frac{v_y}{v_x} + l_{\mathrm{r}}\frac{r}{v_x}\right) - \frac{1}{T_{\mathrm{r}}}F_{\mathrm{r}},$$

where δ_{f} is the front-axle steering angle; c_{f} and c_{r} are the front- and rear-axle cornering stiffnesses; and T_{r} is a speed-dependent tire relaxation constant (see, e.g., [22]). In modern cars with electronic stability control, the main measurements that describe the lateral dynamics are the yaw rate r and the lateral acceleration $a_y = \frac{1}{m}(F_{\mathrm{f}} + F_{\mathrm{r}})$. Considering δ_{f} as the input, the system is described by

$$\hat{A} = \begin{bmatrix} 0 & -v_x & \frac{1}{m} & \frac{1}{m} \\ 0 & 0 & \frac{l_{\mathrm{f}}}{J} & -\frac{l_{\mathrm{r}}}{J} \\ -\frac{c_{\mathrm{f}}}{T_{\mathrm{r}}v_x} & -\frac{l_{\mathrm{f}}c_{\mathrm{f}}}{T_{\mathrm{r}}v_x} & -\frac{1}{T_{\mathrm{r}}} & 0 \\ -\frac{c_{\mathrm{r}}}{T_{\mathrm{r}}v_x} & \frac{l_{\mathrm{r}}c_{\mathrm{r}}}{T_{\mathrm{r}}v_x} & 0 & -\frac{1}{T_{\mathrm{r}}} \end{bmatrix}, \qquad \hat{B} = \begin{bmatrix} 0 \\ 0 \\ \frac{c_{\mathrm{f}}}{T_{\mathrm{r}}} \\ 0 \end{bmatrix},$$
$$\hat{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & \frac{1}{m} \end{bmatrix}, \qquad \hat{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

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

$$A = \begin{bmatrix} -\frac{1}{T_{\mathrm{r}}} & 1 & 0 & \frac{T_{\mathrm{r}}l_{\mathrm{f}}m}{c_{\mathrm{r}}(l_{\mathrm{f}}+l_{\mathrm{r}})} \\ -\frac{l_{\mathrm{r}}c_{\mathrm{r}}(l_{\mathrm{f}}+l_{\mathrm{r}})}{v_x T_{\mathrm{r}} J} & 0 & 1 & \frac{l_{\mathrm{f}}m}{c_{\mathrm{r}}(l_{\mathrm{f}}+l_{\mathrm{r}})} \\ -\frac{c_{\mathrm{r}}(l_{\mathrm{f}}+l_{\mathrm{r}})}{T_{\mathrm{r}}J} & 0 & 0 & \frac{1}{v_x} \\ \frac{c_{\mathrm{r}}(l_{\mathrm{r}}c_{\mathrm{r}}-l_{\mathrm{f}}c_{\mathrm{f}})(l_{\mathrm{f}}+l_{\mathrm{r}})}{mT_{\mathrm{r}}^2 v_x J} & 0 & -\frac{c_{\mathrm{f}}+c_{\mathrm{r}}}{mT_{\mathrm{r}}} & -\frac{1}{T_{\mathrm{r}}} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and the transformations

$$\Gamma_1 = \begin{bmatrix} 0 & 0 & v_x & 0 \\ \frac{c_{\mathrm{r}}(l_{\mathrm{f}}+l_{\mathrm{r}})}{T_{\mathrm{r}}J} & 0 & 0 & 0 \\ -\frac{c_{\mathrm{r}}}{T_{\mathrm{r}}^2} & \frac{c_{\mathrm{r}}}{T_{\mathrm{r}}} & 0 & m \\ \frac{c_{\mathrm{r}}}{T_{\mathrm{r}}^2} & -\frac{c_{\mathrm{r}}}{T_{\mathrm{r}}} & 0 & 0 \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} 0 & \frac{c_{\mathrm{r}}(l_{\mathrm{f}}+c_{\mathrm{r}})}{T_{\mathrm{r}}J} \\ 1 & 0 \end{bmatrix}, \quad \Gamma_3 = \frac{mT_{\mathrm{r}}}{c_{\mathrm{f}}}.$$

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

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

$$A = \begin{bmatrix} 0 & 1 & -v_x & 0 \\ -\frac{c_\mathrm{f}+c_\mathrm{r}}{mT_\mathrm{r}v_x} & -\frac{1}{T_\mathrm{r}} & \frac{l_\mathrm{r}c_\mathrm{r}-l_\mathrm{f}c_\mathrm{f}}{mT_\mathrm{r}v_x} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{l_\mathrm{r}c_\mathrm{r}-l_\mathrm{f}c_\mathrm{f}}{JT_\mathrm{r}v_x} & 0 & -\frac{l_\mathrm{f}^2c_\mathrm{f}+l_\mathrm{r}^2c_\mathrm{r}}{JT_\mathrm{r}v_x} & -\frac{1}{T_\mathrm{r}} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

and the transformations

$$\Gamma_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{l_\mathrm{r}m}{l_\mathrm{f}+l_\mathrm{r}} & 0 & \frac{J}{l_\mathrm{f}+l_\mathrm{r}} \\ 0 & \frac{l_\mathrm{f}m}{l_\mathrm{f}+l_\mathrm{r}} & 0 & -\frac{J}{l_\mathrm{f}+l_\mathrm{r}} \end{bmatrix}, \quad \Gamma_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \Gamma_3 = \begin{bmatrix} \frac{l_\mathrm{r}T_\mathrm{r}m}{c_\mathrm{f}(l_\mathrm{f}+l_\mathrm{r})} & \frac{T_\mathrm{r}J}{c_\mathrm{f}(l_\mathrm{f}+l_\mathrm{r})} \\ \frac{l_\mathrm{f}T_\mathrm{r}m}{c_\mathrm{r}(l_\mathrm{f}+l_\mathrm{r})} & -\frac{T_\mathrm{r}J}{c_\mathrm{r}(l_\mathrm{f}+l_\mathrm{r})} \end{bmatrix},$$

with dimensions $1, 0, 0, 3$. This means that the first state of the system belongs to the zero dynamics subsystem $\boldsymbol{x_a}$, and the remaining three states belong to the $\boldsymbol{x_d}$ subsystem. The $\boldsymbol{x_d}$ subsystem consists of two integrator chains; one of dimension one, and one of dimension two. We conclude that the system is invertible, due to the lack of $\boldsymbol{x_b}$ and $\boldsymbol{x_c}$ subsystems. The A_{aa} matrix is identically 0, meaning that the system has a zero at the origin. Hence, the relationship between the steering angle inputs and the yaw rate and lateral acceleration outputs is non-minimum phase.

Referring back to our discussion of geometry theory, we see that the weakly unobservable subspace is spanned by the vector $[1, 0, 0, 0]^\mathsf{T}$. Transformed back to the original coordinate basis, this corresponds to the state v_y. We therefore know that a hypothetical disturbance occurring in $\dot{v}_y$ can be decoupled from the outputs a_y and r by state feedback (and the SCB representation tells us exactly how to do it). However, we also know that the resulting subsystem would not be asymptotically stable, since the non-minimum phase zero would become a pole of the closed-loop system.

5.2 DC Motor with Friction

According to [23], a DC motor process can be described by the equations

$$\dot{\Omega} = \omega,$$
$$J\dot{\omega} = u - F,$$

where Ω is the shaft angular position, ω is the angular rate, u is the DC motor torque, F is a friction torque, and $J = 0.0023\,\mathrm{kg\,m}^2$ is the motor and load inertia. The friction torque can be modeled by the dynamic LuGre friction model

$$F = \sigma_0 z + \sigma_1 \dot{z} + \alpha_2 \omega,$$
$$\dot{z} = \omega - \frac{\sigma_0 z|\omega|}{\zeta(\omega)},$$

where $\zeta(\omega) = \alpha_0 + \alpha_1 \exp(-(\omega/\omega_0)^2)$. Numerical values for the friction parameters are $\sigma_0 = 260.0\,\mathrm{Nm/rad}$, $\sigma_1 = 0.6\,\mathrm{Nm\,s/rad}$, $\alpha_0 = 0.28\,\mathrm{Nm}$, $\alpha_1 = 0.05\,\mathrm{Nm}$, $\alpha_2 = 0.0176\,\mathrm{Nm\,s/rad}$, and $\omega_0 = 0.01\,\mathrm{rad/s}$. The system can be viewed as consisting of a linear part with a nonlinear perturbation $\sigma_0 z|\omega|/\zeta(\omega)$. Assuming that only the shaft position Ω is measured, a nonlinear observer can be designed for this system by using the time scale assignment techniques from [17]. To do so, it is necessary to find the SCB form of the system, with the nonlinear perturbation $\sigma_0 z|\omega|/\zeta(\omega)$ considered as the sole input. The original system with the nonlinear perturbation as the input is described by the matrices

$$\hat{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{J}(\alpha_2 + \sigma_1) & -\frac{1}{J}\sigma_0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \hat{B} = \begin{bmatrix} 0 \\ \frac{1}{J}\sigma_1 \\ -1 \end{bmatrix}, \tag{8a}$$

$$\hat{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad \hat{D} = 0. \tag{8b}$$

Inserting numerical values and using the Linear Systems Toolkit [12] yields the SCB matrices

$$A \approx \begin{bmatrix} -433.3 & -707.1 & 0 \\ 0 & 0 & 1 \\ -1.1 \cdot 10^5 & -1.8 \cdot 10^5 & 164.8 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \qquad D = 0,$$

where the first state belongs to the zero dynamics subsystem $\boldsymbol{x_a}$, and the remaining two states consist of an integrator chain of length two, in the $\boldsymbol{x_d}$ subsystem. As suggested by the large elements in the system matrices, the problem is poorly conditioned, and we find that we require very large gains to stabilize the system. Using our Maple procedure, we obtain the SCB matrices

$$A = \begin{bmatrix} -\frac{\sigma_0}{\sigma_1} & -\frac{\sigma_0(\sigma_0 J - \sigma_1\alpha_2)}{\sigma_1^3} & 0 \\ 0 & 0 & 1 \\ -\frac{\sigma_0}{J} & -\frac{\sigma_0(\sigma_0 J - \sigma_1\alpha_2)}{J\sigma_1^2} & \frac{\sigma_0 J - \sigma_1\alpha_2 - \sigma_1^2}{J\sigma_1} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \qquad D = 0.$$

This reveals that the small parameter σ_1 generates a singularity for several elements of A, even though the original matrices in (8) did not have any singularities with respect to this parameter. In particular, we see that σ_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 [24]. Using the approximation $\sigma_1 = 0$ results in a dramatically different structure, with the SCB consisting of a single integrator chain of length three, represented by the SCB matrices

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{\sigma_0}{J} & -\frac{\alpha_2}{J} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad D = 0.$$

Proceeding with the observer gain selection based on this system, we obtain good results without using high gains.

5.3 Tenth-Order System

Our last example is a strictly proper, tenth-order system from [1]:

$$
\hat{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \hat{C}^{\mathsf{T}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.
$$

The Maple procedure gives the SCB system matrices

$$
A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ 2 & -12 & 0 & 2 & -8 & 8 & 0 & 8 & 0 & 0 \\ 2 & -4 & -2 & \frac{1}{2} & -2 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & -2 & -2 & 0 & 0 & -1 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & -\frac{1}{2} & 2 & -2 & 0 & -2 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, C^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},
$$

and the dimensions $1, 2, 1, 6$. Hence, the first state belongs to the $\boldsymbol{x_a}$ subsystem, and we can therefore easily see that the system has a non-minimum phase invariant zero at 1. The next two states belong to the $\boldsymbol{x_b}$ subsystem; thus, the system is not right-invertible. The fourth state belongs to the $\boldsymbol{x_c}$ subsystem; thus, the system is not left-invertible. Finally, the last six states consists of three integrator chains of length 1, 2, and 3, respectively, belonging to the $\boldsymbol{x_d}$ subsystem.

6 Numerical Issues

The procedure described in this paper uses exact operations only; thus, there is no uncertainty in the results produced by the decomposition algorithm. The algorithm is primarily based on rank operations and the construction of bases for various subspaces. Rank operations are discontinuous, in the sense that arbitrarily small perturbations to a matrix may alter its rank. This implies that, when a decomposition is carried out using exact operations, arbitrarily small perturbations to system matrices may fundamentally alter the identified structure of a system. This is in contrast to decompositions based on floating-point operations, which may be insensitive to small perturbations to the system matrices.

Whether exactness is desirable or not depends on the application. When the input data is exact or the system model is based on first principles, an exact decomposition may help to reveal fundamental structural properties of the system and how these properties are affected by various quantities in the system matrices. If, on the other hand, the system matrices have been derived based on experimental system identification, an exact decomposition may not be desirable, and it may even provide misleading information about the system structure. Thus, the exact procedure presented here is not a replacement for numerical tools developed for the same purpose.

Throughout the decomposition algorithm, a number of non-unique transformation matrices must be constructed. In the Maple procedure, these matrices are constructed with the goal of having a simple structure, based on the assumption that fewer changes to the original states will result in less complicated symbolic expressions in the computed SCB system. Depending on the structure and dimensions of the system, however, the procedure may still result in complicated expressions, and if the original system matrices contain complicated expressions, these will in general not be simplified.

A precise analysis of the computational complexity of the procedure is difficult, due to the complex nature of the decomposition algorithm and the underlying Maple functions. However, it is possible to make some practical observations regarding this issue. Executed in Maple 12 on an Intel Pentium processor with two 2-MHz cores, the total CPU time needed for decomposition of the single-track model was approximately 0.30 s for the single-input case and 0.21 s for the double-input case. For the DC motor example, the total CPU time was approximately 0.19 s for the original matrices in (8) and 0.25 s with the modification $\sigma_1 = 0$. For the tenth-order example, the total CPU time was approximately 0.48 s. These execution times illustrate that an increase in the order of the system does not automatically result in a large increase in execution time; the structure of the system and the complexity of the expressions in the system matrices has a greater impact on execution time. For example, randomly generated, strictly proper systems with 20 states, 4 inputs and 4 outputs, with the system matrices made up of integers between -10 and 10 with 25% density, are generally decomposed in less than 0.4 s. If, on the other hand, the number of inputs is reduced to 3, the decomposition generally takes around 50 s. The reason for this large difference is that, in the former case, the computed SCB systems generally consist of an $\boldsymbol{x_a}$ subsystem with 16 states and an $\boldsymbol{x_d}$ subsystem with four states, which requires only a single iteration of the algorithm for identifying $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$ (described at the beginning of Section 4.1). In the latter case, the computed SCB systems generally consist of an $\boldsymbol{x_b}$ subsystem with 17 states, and an $\boldsymbol{x_d}$ subsystem with three states, which requires 17 increasingly complex iterations of the algorithm for identifying $\boldsymbol{x_b}$ and $\boldsymbol{x_d}$.

7 Concluding Remarks

We have presented a procedure written in the mathematics software suite Maple, which is capable of transforming any linear time-invariant system described by exact symbols and fractions to the SCB, and we have illustrated the use of this algorithm on several examples.

The DC motor 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 such as squaring down of non-square systems and asymptotic time scale assignment.

Acknowledgments

The work of Håvard Fjær Grip is supported by the Research Council of Norway. The work of Ali Saberi is partially supported by NAVY grants ONR KKK777SB001 and ONR KKK760SB0012.

References

[1] P. Sannuti and A. Saberi, "Special coordinate basis for multivariable linear systems—finite and infinite zero structure, squaring down and decoupling," *Int. J. Contr.*, vol. 45, no. 5, pp. 1655–1704, 1987.

[2] A. Saberi, B. M. Chen, and P. Sannuti, *Loop Transfer Recovery: Analysis and Design*. London: Springer, 1993.

[3] A. Saberi, P. Sannuti, and B. M. Chen, *H_2 Optimal Control*. Prentice Hall, 1995.

[4] B. M. Chen, *Robust and H_∞ Control*. London: Springer, 2000.

[5] B. M. Chen, Z. Lin, and Y. Shamash, *Linear Systems Theory: A Structural Decomposition Approach*. Boston: Birkhäuser, 2004.

[6] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Filtering Theory*. Boston: Birkhäuser, 2006.

[7] B. M. Chen, A. Saberi, and P. Sannuti, "Explicit expressions for cascade factorization of general nonminimum phase systems," *IEEE Trans. Automat. Contr.*, vol. 37, no. 3, pp. 358–363, 1992.

[8] A. Saberi and P. Sannuti, "Squaring down of non-strictly proper systems," *Int. J. Contr.*, vol. 51, no. 3, pp. 621–629, 1990.

[9] H. K. Ozcetin, A. Saberi, and P. Sannuti, "Special coordinate basis for order reduction of linear multivariable systems," *Int. J. Contr.*, vol. 52, no. 1, pp. 191–226, 1990.

[10] L. M. Silverman, "Inversion of multivariable linear systems," *IEEE Trans. Automat. Contr.*, vol. 14, no. 3, pp. 270–276, 1969.

[11] D. Chu, X. Liu, and R. C. E. Tan, "On the numerical computation of a structural decomposition in systems and control," *IEEE Trans. Automat. Contr.*, vol. 47, no. 11, pp. 1786–1799, 2002.

[12] X. Liu, B. M. Chen, and Z. Lin, "Linear systems toolkit in Matlab: structural decompositions and their applications," *J. Contr. Theor. Appl.*, vol. 3, pp. 287–294, 2005.

[13] A. Saberi, B. M. Chen, and P. Sannuti, "Theory of LTR for non-minimum phase systems, recoverable target loops, and recovery in a subspace Part 1. Analysis," *Int. J. Contr.*, vol. 53, no. 5, pp. 1067–1115, 1991.

[14] M. C. Berg, "Introduction to a special coordinate basis for multivariable linear systems," *IEE Proc. Contr. Theor. Appl.*, vol. 145, no. 2, pp. 204–210, 1998.

[15] A. S. Morse, "Structural invariants of linear multivariable systems," *SIAM J. Contr.*, vol. 11, no. 3, 1973.

[16] A. G. J. MacFarlane and N. Karcanias, "Poles and zeros of linear multivariable systems: a survey of the algebraic, geometric and complex-variable theory," *Int. J. Contr.*, vol. 24, no. 1, pp. 33–74, 1976.

[17] A. Saberi and P. Sannuti, "Observer design for loop transfer recovery and for uncertain dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 35, no. 8, pp. 878–897, 1990.

[18] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed. Springer-Verlag, 1979.

[19] H. L. Trentelman, A. A. Stoorvogel, and M. Hautus, *Control Theory for Linear Systems*. Springer, 2001.

[20] H. F. Grip and A. Saberi, "Maple source code for structural decomposition of linear multivariable systems, version 0.2," 2010, available from: http://www.itk.ntnu.no/ansatte/Grip_Havard.Fjar/sw/scb.mpl.

[21] U. Kiencke and L. Nielsen, *Automotive Control Systems*. Springer, 2000.

[22] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Butterworth-Heinemann, 2006.

[23] C. Canudas de Wit and P. Lischinsky, "Adaptive friction compensation with partially known dynamic friction model," *Int. J. Adapt. Contr. Signal Process.*, vol. 11, no. 1, pp. 65–80, 1997.

[24] S. Sastry, J. Hauser, and P. Kokotovic, "Zero dynamics of regularly perturbed systems may be singularly perturbed," *Syst. Contr. Lett.*, vol. 13, no. 4, pp. 299–314, 1989.