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 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_∞ 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

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}\hat{u}, \quad \hat{y} = \hat{C}\hat{x} + \hat{D}\hat{u}. \tag{1}$$

where $\hat{x} \in \mathbb{R}^n$ is the state, $\hat{u} \in \mathbb{R}^m$ is the input, and $\hat{y} \in \mathbb{R}^p$ 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 $\hat{u} = [u_0^\mathsf{T}, \hat{u}_1^\mathsf{T}]^\mathsf{T}$, and $\hat{y} = [y_0^\mathsf{T}, \hat{y}_1^\mathsf{T}]^\mathsf{T}$, where u_0 and 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

$$\hat{y} = \begin{bmatrix} y_0 \\ \hat{y}_1 \end{bmatrix} = \begin{bmatrix} C_0\hat{x} + u_0 \\ \hat{C}_1\hat{x} \end{bmatrix}, \tag{2}$$

Håvard Fjær Grip is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. Ali Saberi is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164-2752.

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, and National Aeronautics and Space Administration grant NNA06CN26A.

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 $u_0 = y_0 - C_0\hat{x}$, we can write the system (1) in the alternative form

$$\dot{\hat{x}} = (\hat{A} - B_0C_0)\hat{x} + \hat{B}\begin{bmatrix} y_0^\mathsf{T} & \hat{u}_1^\mathsf{T} \end{bmatrix}^\mathsf{T}, \quad \hat{y} = \hat{C}\hat{x} + \hat{D}\hat{u}. \tag{3}$$

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 Γ_1, Γ_2, and Γ_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 $x = \mathrm{col}(x_a, x_b, x_c, x_d)$, where each component represents a particular subsystem described in the next section. The output is partitioned as $y = \mathrm{col}(y_0, y_d, y_b)$, 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 $u = \mathrm{col}(u_0, u_d, u_c)$, 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, Γ_3 is on the form $\mathrm{diag}(I_{m_0}, \bar{\Gamma}_3)$, for some nonsingular $\bar{\Gamma}_3$.

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 effect 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

$$\dot{x}_a = A_{aa}x_a + B_{a0}y_0 + L_{ad}y_d + L_{ab}y_b, \tag{4a}$$
$$\dot{x}_b = A_{bb}x_b + B_{b0}y_0 + L_{bd}y_d, \tag{4b}$$
$$\dot{x}_c = A_{cc}x_c + B_{c0}y_0 + L_{cd}y_d + L_{cb}y_b + A_{ca}x_a + B_cu_c, \tag{4c}$$
$$\dot{x}_i = A_{q_i}x_i + B_{d0}y_0 + L_{id}y_d + B_{q_i}(u_i + E_{ia}x_a + E_{ib}x_b + E_{ic}x_c + E_{id}x_d), \tag{4d}$$

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

$$y_0 = C_{0a}x_a + C_{0b}x_b + C_{0c}x_c + C_{0d}x_d + u_0, \tag{5a}$$
$$y_i = C_{q_i}x_i, \quad i = 1, \ldots, m_d, \tag{5b}$$
$$y_b = C_bx_b. \tag{5c}$$

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

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

The pair (A_{bb}, C_b) is observable, and the pair (A_{cc}, B_c) is controllable. In the strictly proper case, the input u_0 and output 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}.

C. Compact Form

We may write (4) as

$$\dot{x} = Ax + B\begin{bmatrix} y_0^\mathsf{T} & u_d^\mathsf{T} & u_c^\mathsf{T} \end{bmatrix}^\mathsf{T}, \quad y = Cx + Du, \tag{6}$$

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

$$A = \begin{bmatrix} A_{aa} & L_{ab}C_b & 0 & L_{ad}C_d \\ 0 & A_{bb} & 0 & L_{bd}C_d \\ A_{ca} & L_{cb}C_b & A_{cc} & L_{cd}C_d \\ B_dE_{da} & B_dE_{db} & B_dE_{dc} & A_{dd} \end{bmatrix} \quad B = \begin{bmatrix} B_{a0} & 0 & 0 \\ B_{b0} & 0 & 0 \\ B_{c0} & 0 & B_c \\ B_{d0} & B_d & 0 \end{bmatrix}$$

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

where $A_{dd} = \mathrm{diag}(A_{q_1}, \ldots, A_{q_{m_d}}) + L_{dd}C_d + B_dE_{dd}$, $B_d = \mathrm{diag}(B_{q_1}, \ldots, B_{q_{m_d}})$, $C_d = \mathrm{diag}(C_{q_1}, \ldots, C_{q_{m_d}})$, $L_{dd} = [L_{1d}^\mathsf{T}, \ldots, L_{m_dd}^\mathsf{T}]^\mathsf{T}$, $E_{da} = [E_{1a}^\mathsf{T}, \ldots, E_{m_da}^\mathsf{T}]^\mathsf{T}$, 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 $\hat{D}$ 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 Γ_3 is of the form $\mathrm{diag}(I_{m_0}, \bar{\Gamma}_3)$, we can make the substitution $\mathrm{col}(y_0, \hat{u}_1) = \Gamma_3 \mathrm{col}(y_0, u_d, u_c)$. We then obtain the equations

$$\dot{x} = \Gamma_1^{-1}(\hat{A} - B_0C_0)\Gamma_1 x + \Gamma_1^{-1}\hat{B}\Gamma_3\begin{bmatrix} y_0^\mathsf{T} & u_d^\mathsf{T} & u_c^\mathsf{T} \end{bmatrix}^\mathsf{T},$$

$$y = \Gamma_2^{-1}\hat{C}\Gamma_1 x + \Gamma_2^{-1}\hat{D}\Gamma_3 u.$$

Comparison with (6) then shows that $A = \Gamma_1^{-1}(\hat{A} - B_0C_0)\Gamma_1$, $B = \Gamma_1^{-1}\hat{B}\Gamma_3$, $C = \Gamma_2^{-1}\hat{C}\Gamma_1$, 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 $\tilde{u}$, output $\tilde{y}$, input matrix $\tilde{B}$, and output matrices $\tilde{C}$ and $\tilde{D}$. Then there are nonsingular transformations U and Y such that $\tilde{u} = U\hat{u}$ and $\tilde{y} = Y\hat{y}$, where $\hat{u}$ and $\hat{y}$ have the structure required in (2). The dimension m_0 of u_0 and 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).

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_{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. (ii) The system (1) is right-invertible if, and only if, the subsystem x_b is non-existent. (iii) The system (1) is left-invertible if, and only if, the subsystem x_c is non-existent. 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, 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), as described in Section II-D. Finally, the output dim is a list of four integers representing the dimensions of the x_a, x_b, x_c, 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_{i0}, given by the expression $y_{i0} = C_i\hat{x}$, 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.

A. 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 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 **scbSP** procedure be represented by the state equations $\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}\hat{u}$, $\hat{y} = \hat{C}\hat{x}$. In the first iteration we start with the output $y_{10} = \hat{C}\hat{x}$, and determine whether its derivative $\dot{y}_{10} = \hat{C}\hat{A}\hat{x} + \hat{C}\hat{B}\hat{u}$ depends on any part of the input $\hat{u}$. 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 x_d subsystem. A transformed part of the output derivative that is not directly influenced by the input is denoted $\tilde{C}_1\hat{x}$, and is processed further. We use a transformation $\bar{\phi}_1$ to separate out any part of $\tilde{C}_1\hat{x}$ 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 $\bar{S}_i$ 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 $\bar{\mathfrak{q}}_{i-1} \times m$ of maximal rank $\bar{\mathfrak{q}}_{i-1}$, let $\bar{\mathfrak{q}}_i$ 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

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

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 are 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** procedure 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** procedure 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** 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_{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 v such that $v[\bar{D}_{i-1}^{\mathsf{T}}, (C_i\hat{B})^{\mathsf{T}}]^{\mathsf{T}} = 0$, where the rightmost p_i columns of v are zero. However, since the rows of $\bar{D}_{i-1}$ are linearly independent, this implies that $v = 0$, 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 x_b and x_d. After x_b and x_d are identified, finding the input and output transformations Γ_3 and Γ_2 is straightforward, based on [1].

B. Constructing the x_a and x_c States

After finding the transformations from the original states to the x_b and x_d states, and the transformations Γ_3 and Γ_2, the next step is to find a transformation to a temporary state vector x_s that will be further decomposed into the states x_a and x_c. The requirements on x_s is that it 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_a and x_c in the strictly proper case.

Suppose that $\mathrm{col}(x_b, x_d) = \Gamma_{bd}\hat{x}$. The procedure for finding x_s is to start with a temporary state vector $x_s^0 = \Gamma_s^0\hat{x}$ that is linearly independent of x_b and 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 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_d, can be written as

$$\begin{aligned}\dot{x}_s^0 &= A^0 \begin{bmatrix} x_s^{0\mathsf{T}} & x_b^{\mathsf{T}} & x_d^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} + B^0 \begin{bmatrix} u_d^{\mathsf{T}} & u_c^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \\ &= A_s^0 x_s^0 + A_b^0 x_b + A_d^0 x_d + B_d^0 u_d + B_c^0 u_c,\end{aligned}$$

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** procedure of the LinearAlgebra package.

To conform with the SCB, we need to modify x_s^0 to eliminate the input u_d in $\dot{x}_s^0$. To eliminate u_d, we create a temporary state vector $x_{d0} = \Gamma_{d0}\hat{x}$, 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 $\dot{x}_{d0} = u_d + A_{d0}[x_s^{0\mathsf{T}}, x_b^{\mathsf{T}}, x_d^{\mathsf{T}}]^{\mathsf{T}}$, for some matrix A_{d0}. Therefore, by defining a new temporary state $x_s^1 = x_s^0 - B_d^0 x_{d0}$, we have $\dot{x}_s^1 = (A^0 - B_d^0 A_{d0})[x_s^{0\mathsf{T}}, x_b^{\mathsf{T}}, x_d^{\mathsf{T}}] + B_c^0 u_c$. Hence, the derivative of the new temporary state vector x_s^1 is independent of u_d, 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 $\dot{x}_s$ depends only on x_s, y_b, y_d, and u_c.

The final step is to decompose x_s into a subsystem x_a that is unaffected by the input u_c, and a subsystem x_c that is controllable from u_c. We do this by transforming x_s to the Kalman controllable canonical form. First, we find the derivative $\dot{x}_s = A_{ss}x_s + A_{sb}x_b + A_{sd}x_d + B_{sc}u_c$, for some matrices A_{ss}, A_{sb}, A_{sd}, and B_{sc}. We then compute the controllability matrix $C_{\mathrm{ctr}} = [B_{sc}, A_{ss}B_{sc}, \ldots, A_{ss}^{n_s-1}B_{sc}]$, where n_s is the dimension of x_s. To transform x_s to the Kalman controllable canonical form, we define $\mathrm{col}(x_a, x_c) = [T_2, T_1]^{-1}x_s$,

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_1 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 Γ_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 $\hat{A}$, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$. We need to find nonsingular U and Y such that, according to Section II-D, $\hat{B} = \tilde{B}U$, $\hat{C} = Y^{-1}\tilde{C}$, and $\hat{D} = Y^{-1}\tilde{D}U$, where $\hat{D}$ is of the form $\text{diag}(I_{m_0}, 0)$. The rank m_0 of $\tilde{D}$ is found using the **Rank** procedure. 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 $Y^{-1}\tilde{D}U = [(Y_1\tilde{D}U)^{\mathsf{T}}, (Y_2\tilde{D}U)^{\mathsf{T}}]^{\mathsf{T}}$, where $Y_1\tilde{D}U = [I_{m_0}, 0]$ and $Y_2\tilde{D}U = 0$. To solve these equations, we choose the rows of Y_2 from the left null space of $\tilde{D}$, using **NullSpace** and **ReducedRowEchelonForm** as before; and we select Y_1 such that $[Y_1^{\mathsf{T}}, Y_2^{\mathsf{T}}]^{\mathsf{T}}$ is nonsingular, using the same procedure as for finding S_{i1} given S_{i2} in Section III-A.1. This leaves us to solve the equation $Y_1\tilde{D}U = [I_{m_0}, 0]$ with respect to some nonsingular U. Let $U^{-1} = [U_1^{\mathsf{T}}, U_2^{\mathsf{T}}]^{\mathsf{T}}$ such that U_1 has m_0 rows. We select $U_1 = Y_1\tilde{D}$, and we select U_2 such that $[U_1^{\mathsf{T}}, U_2^{\mathsf{T}}]^{\mathsf{T}}$ 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).

Let 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

$$\dot{\hat{x}} = (\hat{A} - B_0C_0)\hat{x} + B_0y_0 + \hat{B}_1\hat{u}_1, \tag{7a}$$

$$y_0 = C_0\hat{x} + u_0, \quad \hat{y}_1 = \hat{C}_1\hat{x}. \tag{7b}$$

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 $\bar{\Gamma}_1$, $\bar{\Gamma}_2$, and $\bar{\Gamma}_3$. Substituting $\hat{x} = \bar{\Gamma}_1 x$, $\hat{y}_1 = \bar{\Gamma}_2[y_d^{\mathsf{T}}, y_b^{\mathsf{T}}]^{\mathsf{T}}$, and $\hat{u}_1 = \bar{\Gamma}_3[u_d^{\mathsf{T}}, u_c^{\mathsf{T}}]^{\mathsf{T}}$ in (7) yields

$$\dot{x} = \bar{\Gamma}_1^{-1}(\hat{A} - B_0C_0)\bar{\Gamma}_1 x + \bar{\Gamma}_1^{-1}B_0y_0 + \bar{\Gamma}_1^{-1}\hat{B}_1\bar{\Gamma}_3 \begin{bmatrix} u_d^{\mathsf{T}} & u_c^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}},$$

$$y_0 = C_0\bar{\Gamma}_1 x + u_0, \quad \begin{bmatrix} y_d^{\mathsf{T}} & y_b^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} = \bar{\Gamma}_2^{-1}\hat{C}_1\bar{\Gamma}_1 x.$$

It is easily confirmed that this system conforms to the SCB, by defining $A = \bar{\Gamma}_1^{-1}(\hat{A} - B_0C_0)\bar{\Gamma}_1$, $B = \bar{\Gamma}_1^{-1}[B_0, \hat{B}_1\bar{\Gamma}_3]$, $C = [C_0^{\mathsf{T}}, (\bar{\Gamma}_2^{-1}\hat{C}_1)^{\mathsf{T}}]^{\mathsf{T}}\bar{\Gamma}_1$, and $D = \text{diag}(I_{m_0}, 0)$. Defining the transformations for the non-strictly proper system as $\Gamma_1 = \bar{\Gamma}_1$, $\Gamma_2 = \text{diag}(I_{m_0}, \bar{\Gamma}_2)$, and $\Gamma_3 = \text{diag}(I_{m_0}, \bar{\Gamma}_3)$, we obtain $A = \Gamma_1^{-1}(\hat{A} - B_0C_0)\Gamma_1$, $B = \Gamma_1^{-1}\hat{B}\Gamma_3$, $C = \Gamma_2^{-1}\hat{C}\Gamma_1$, 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 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 $\dot{v}_y = \frac{1}{m}(F_{\text{f}} + F_{\text{r}}) - rv_x$, $\dot{r} = \frac{1}{J}(l_{\text{f}}F_{\text{f}} - l_{\text{r}}F_{\text{r}})$, 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_{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 $\dot{F}_{\text{f}} = \frac{c_{\text{f}}}{T_{\text{r}}}(\delta_{\text{f}} - \frac{v_y}{v_x} - l_{\text{f}}\frac{r}{v_x}) - \frac{1}{T_{\text{r}}}F_{\text{f}}$ and $\dot{F}_{\text{r}} = \frac{c_{\text{r}}}{T_{\text{r}}}(-\frac{v_y}{v_x} + l_{\text{r}}\frac{r}{v_x}) - \frac{1}{T_{\text{r}}}F_{\text{r}}$, 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., [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 $a_y = \frac{1}{m}(F_{\text{f}} + F_{\text{r}})$. Considering δ_{f} as the input, the system is described by

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

$$\hat{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & \frac{1}{m} \end{bmatrix}, \qquad \hat{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

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

$$A = \begin{bmatrix} -\frac{1}{T_{\text{r}}} & 1 & 0 & \frac{T_{\text{r}}l_{\text{f}}m}{c_{\text{r}}(l_{\text{f}}+l_{\text{r}})} \\ -\frac{l_{\text{r}}c_{\text{r}}(l_{\text{f}}+l_{\text{r}})}{v_xT_{\text{r}}J} & 0 & 1 & \frac{l_{\text{f}}m}{c_{\text{r}}(l_{\text{f}}+l_{\text{r}})} \\ -\frac{c_{\text{r}}(l_{\text{f}}+l_{\text{r}})}{T_{\text{r}}J} & 0 & 0 & \frac{1}{v_x} \\ \frac{c_{\text{r}}(l_{\text{r}}c_{\text{r}}-l_{\text{f}}c_{\text{f}})(l_{\text{f}}+l_{\text{r}})}{mT_{\text{r}}^2v_xJ} & 0 & -\frac{c_{\text{f}}+c_{\text{r}}}{mT_{\text{r}}} & -\frac{1}{T_{\text{r}}} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

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

The dimension list dim returned by the procedure is $0, 3, 0, 1$, meaning that the first three states belong to the x_b 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_b and x_d subsystems are always observable. The system is left-invertible, since the state x_c 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_b 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_a.

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

SCB system matrices

$$A = \begin{bmatrix} 0 & 1 & -v_x & 0 \\ -\frac{c_f+c_r}{mT_r v_x} & -\frac{1}{T_r} & \frac{l_r c_r - l_f c_f}{mT_r v_x} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{l_r c_r - l_f c_f}{JT_r v_x} & 0 & \frac{l_f^2 c_f + l_r^2 c_r}{JT_r v_x} & -\frac{1}{T_r} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

with dimensions $1,0,0,3$. This means that the first state of the system belongs to the zero dynamics x_a, 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 $A_{aa} = 0$, 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 $\dot{\Omega} = \omega$, $J\dot{\omega} = u - F$, 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 $F = \sigma_0 z + \sigma_1 \dot{z} + \alpha_2 \omega$, $\dot{z} = \omega - \sigma_0 z|\omega|/\zeta(\omega)$, where $\zeta(\omega) = \alpha_0 + \alpha_1 \exp(-(\omega/\omega_0)^2)$. Numerical values for the friction parameters are $\sigma_0 = 260.0\,\mathrm{Nm/rad}$, $\sigma_1 = 0.6\,\mathrm{Nm\,s/rad}$, $\alpha_0 = 0.28\,\mathrm{Nm}$, $\alpha_1 = 0.05\,\mathrm{Nm}$, $\alpha_2 = 0.176\,\mathrm{Nm\,s/rad}$, 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 [16]. 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 is described by the matrices

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

$\hat{C} = [1,0,0]$, and $\hat{D} = 0$. Inserting numerical values and using the Linear Systems Toolkit [9] yields the SCB matrices

$$A \approx \begin{bmatrix} -433.3 & -592.7 & 0 \\ 0 & 0 & 1 \\ -1.1 \cdot 10^5 & -1.5 \cdot 10^5 & 95.9 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$C = [0,1,0]$, and $D = 0$, where the first state belongs to the zero dynamics subsystem x_a, and the remaining two states consist of an integrator chain of length two, in the 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 matrix

$$A = \begin{bmatrix} -\sigma_0/\sigma_1 & -\sigma_0(\sigma_0 J - \sigma_1\alpha_2)/\sigma_1^3 & 0 \\ 0 & 0 & 1 \\ -\sigma_0/J & -\sigma_0(\sigma_0 J - \sigma_1\alpha_2)/(J\sigma_1^2) & (\sigma_0 J - \sigma_1\alpha_2 - \sigma_1^2)/(J\sigma_1) \end{bmatrix}.$$

This reveals that a source of the conditioning problem is powers of the small parameter σ_1 appearing in the denominators, even though it does not appear in any denominators in (8). 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 [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_c 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

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

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

[9] 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.

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

[11] 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.

[12] H. F. Grip and A. Saberi, "Maple source code for structural decomposition of linear multivariable systems, version 0.1," 2009, available from: http://www.itk.ntnu.no/ansatte/Grip_Havard.Fjar/sw/scb_v01.mpl.

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

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

[15] C. Canudas de Wit and P. Lischinsky, "Adaptive friction compensation with partially known dynamic friction model," *Int. J. Adapt. Contr. Signal Process.*, vol. 11, pp. 65–80, 1997.

[16] 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.

[17] 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.