

Technical Notes and Correspondence

Parametrization of the Regular Equivalences of the Canonical Controller

A. Agung Julius, Jan Willem Polderman, and Arjan van der Schaft

Abstract—We study control problems for linear systems in the behavioral framework. Our focus is a class of regular controllers that are equivalent to the canonical controller. The canonical controller is a particular controller that is guaranteed to solve the control problem whenever a solution exists. However, it has been shown that, in most cases, the canonical controller is not regular. The main result of the paper is a parametrization of all regular controllers that are equivalent to the canonical controller. The parametrization is then used to solve two control problems. The first problem is related to designing a regular controller that uses as few control variables as possible. The second problem is to design a regular controller that satisfies a predefined input–output partitioning constraint. In both problems, based on the parametrization, we present algorithms that does the controller design.

Index Terms—Behavior, canonical controller, input–output partition, regularity.

I. INTRODUCTION

In this paper, we discuss control problems for linear differential systems in the behavioral approach. The behavior of the systems discussed in this paper is the set of solutions of the linear differential equations that describe the systems [1]. In particular, we restrict our attention to the class of infinitely differentiable functions \mathcal{C}^∞ . Thus, whenever a differential equation is given, we assume its solution to be infinitely differentiable.

Standard control problems in the behavioral approach to systems theory can be formulated as follows [2]–[4]. A plant to be controlled that has two kinds of variables, *to-be-controlled variables* and *control variables*, is given. Throughout this paper, we denote the control variables by c and the to-be-controlled variables by w . The dimensions of c and w are denoted by c and w , respectively. A behavioral model of the plant system that captures the relevant relation between w and c is called the *full plant behavior*, and is denoted by $\mathcal{P}_{\text{full}}$. The full plant behavior can be compactly represented as the set of all signal pairs (w, c) that are strong solutions to an associated system of linear differential equations [1]

$$\mathcal{P}_{\text{full}} := \left\{ (w, c) \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^{w+c}) \mid R \left(\frac{d}{dt} \right) w + M \left(\frac{d}{dt} \right) c = 0 \right\} \quad (1)$$

where R and M are polynomial matrices of appropriate dimensions. We denote the class of polynomial matrices with indeterminate ξ , g

Manuscript received October 11, 2006; revised August 13, 2007 and September 24, 2007. Recommended by Associate Editor Hansson.

A. A. Julius is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: agung@seas.upenn.edu).

J. W. Polderman is with the Department of Applied Mathematics, University of Twente, Enschede 7500 AE, The Netherlands (e-mail: j.w.polderman@math.utwente.nl).

A. J. van der Schaft is with the Institute for Mathematics and Computer Science, University of Groningen, Groningen 9700 AV, The Netherlands. He is also with the Department of Applied Mathematics, University of Twente, Enschede 7500 AE, The Netherlands (e-mail: A.J.van.der.Schaft@math.rug.nl).

Digital Object Identifier 10.1109/TAC.2008.919515

rows, and q columns over the real field as $\mathbb{R}^{g \times q}[\xi]$. A representation of the behavior in the form of (1) is called a *kernel representation*, the reason being that the behavior is simply the kernel of a linear differential operator.

A *controller* is a device that is attached to (or an algorithm that acts on) the control variables and restricts their behavior. This restriction is imposed on the plant via the control variables, such that it (indirectly) affects the behavior of the to-be-controlled variables. A controller \mathcal{C} is thus a behavior containing all signals c allowed by the controller

$$\mathcal{C} := \left\{ c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \mid C \left(\frac{d}{dt} \right) c = 0 \right\}. \quad (2)$$

The resulting behavior is called the *controlled system*. The *controlled behavior* is then defined as

$$\mathcal{K} := \{ w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \mid \exists c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \text{ such that } (w, c) \in \mathcal{P}_{\text{full}} \text{ and } c \in \mathcal{C} \}. \quad (3)$$

The controlled behavior \mathcal{K} is obtained by eliminating the control variables c from the following kernel representation

$$\begin{bmatrix} R \left(\frac{d}{dt} \right) & M \left(\frac{d}{dt} \right) \\ 0 & C \left(\frac{d}{dt} \right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0. \quad (4)$$

If we eliminate the control variables from the full behavior, we obtain the so-called *manifest behavior*, which is denoted by \mathcal{P} . Thus

$$\mathcal{P} := \{ w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \mid \exists c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \text{ such that } (w, c) \in \mathcal{P}_{\text{full}} \}. \quad (5)$$

As a part of the control problem, one is given a *specification*, which is expressed in terms of the to-be-controlled variables. The specification \mathcal{S} is given by the following kernel representation

$$\mathcal{S} := \left\{ w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \mid S \left(\frac{d}{dt} \right) w = 0 \right\}. \quad (6)$$

The objective of the control problem is to find a controller \mathcal{C} such that $\mathcal{K} = \mathcal{S}$. If such controller exists, then the specification \mathcal{S} is said to be *implementable* and the controller \mathcal{C} is said to implement \mathcal{S} .

In [5] and [6], a particular controller design, called the *canonical controller* was introduced. This design has the nice property that it implements the desired specification if and only if the specification is implementable. However, an analysis on the regularity of the canonical controller reveals that it is *maximally irregular* [7]. Regularity is a desirable property for the interconnection [2], [3], which we will explain in Section II. We show that there exist regular controllers that are equivalent to the canonical controller, and we provide a parametrization of all such controllers. This parametrization is then used to solve the following two control problems.

- 1) The problem of *control with minimal interaction* [8]. This problem is about designing a regular controller that interacts with the plant with as few control variables as possible. The motivation behind this problem is as follows. Consider a situation where the plant and the controller are separated by a large physical distance. We need a communication link between the plant and

the controller to establish the interconnection. It is therefore favorable to have as few control variables as possible, so that the amount of communication links/channels can be minimized.

- 2) The problem of *control with input–output partitioning constraint*. This problem is about designing a regular controller, in which some predetermined control variables remain free in the controller.

II. BACKGROUND MATERIAL

Kernel representations of a given behavior are not unique. Nevertheless, for any behavior \mathfrak{B} , there is a unique integer $p(\mathfrak{B})$, which is the minimum number of rows that a kernel representation of \mathfrak{B} can have. This number is also the row rank of any kernel representation of the behavior. A kernel representation with the minimum number of rows (i.e., equal to its row rank) is called a *minimal kernel representation*.

Suppose that a behavior \mathfrak{B} is given by

$$\mathfrak{B} := \left\{ w \mid R \left(\frac{d}{dt} \right) w = 0 \right\} \quad (7)$$

where R is full row rank and has $p(\mathfrak{B})$ rows. We can permute and partition the variables in w into w_1 and w_2 , such that (7) becomes

$$\mathfrak{B} := \left\{ (w_1, w_2) \mid R_1 \left(\frac{d}{dt} \right) w_1 + R_2 \left(\frac{d}{dt} \right) w_2 = 0 \right\} \quad (8)$$

where R_1 is a square full row rank polynomial matrix. Such a partition is called an *input–output* partition, where w_1 is the output and w_2 is the input to the system. Notice that the number of outputs of \mathfrak{B} is $p(\mathfrak{B})$.

Given a control problem, the implementability of a specification \mathcal{S} is a property that depends on the specification itself as well as the plant. The following result is proven in [9] and [10].

Lemma 1 (Willems' lemma): Given $\mathcal{P}_{\text{full}}$ as a kernel representation of (1). A specification \mathcal{S} is implementable if and only if $\mathcal{N} \subseteq \mathcal{S} \subseteq \mathcal{P}$, where \mathcal{N} is the *hidden behavior* defined by $\mathcal{N} := \{w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \mid (w, 0) \in \mathcal{P}_{\text{full}}\}$.

Quite often, in addition to requiring that the controller implements the desired specification, we also require that the controller possesses a certain property with respect to the plant. A concept that has been quite extensively studied is the so-called *regularity property* [3], [11]–[13]. A controller

$$\mathcal{C} = \left\{ c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \mid C \left(\frac{d}{dt} \right) c = 0 \right\} \quad (9)$$

where C is full row rank, to be regular if

$$\text{rank} \begin{bmatrix} R & M \\ 0 & C \end{bmatrix} = \text{rank} [R \ M] + \text{rank } C. \quad (10)$$

It can be proven that nonregular interconnections might affect the autonomous part of the plant or the controller [2], [14], which, in many cases would be undesirable or unrealistic.

If the specification \mathcal{S} is such that there exists a regular controller \mathcal{C} that implements it, then \mathcal{S} is said to be *regularly implementable*. Necessary and sufficient conditions for regular implementability were derived in [3].

Theorem 2: Given the full plant behavior $\mathcal{P}_{\text{full}}$. A specification \mathcal{S} is regularly implementable if and only if: 1) it is implementable, i.e., $\mathcal{N} \subseteq \mathcal{S} \subseteq \mathcal{P}$ and 2) $\mathcal{S} + \mathcal{P}^{\text{ctr}} = \mathcal{P}$. The symbol \mathcal{P}^{ctr} denotes the controllable part of the manifest behavior \mathcal{P} .

III. CANONICAL CONTROLLER AND ITS REGULAR EQUIVALENCES 122

In this section, we review the idea of canonical controller and its properties [6]. Given a full plant behavior $\mathcal{P}_{\text{full}}$ and a specification \mathcal{S} , the behavior of the canonical controller \mathcal{C}_{can} is defined as 124 125

$$\mathcal{C}_{\text{can}} := \{c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \mid \exists w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \text{ such that } (w, c) \in \mathcal{P}_{\text{full}} \text{ and } w \in \mathcal{S}\}. \quad (11)$$

A kernel representation of the canonical controller can be obtained by eliminating w from the following kernel representation 126 127

$$\begin{bmatrix} R \left(\frac{d}{dt} \right) & M \left(\frac{d}{dt} \right) \\ S \left(\frac{d}{dt} \right) & 0 \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0. \quad (12)$$

For the canonical controller, the following result holds. 128

Theorem 3: (cf. [6]) The canonical controller \mathcal{C}_{can} implements the specification \mathcal{S} if and only if \mathcal{S} is implementable. 129 130

We define the *control manifest behavior* of the plant as 131

$$\mathcal{P}_c := \{c \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^c) \mid \exists w \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^w) \text{ such that } (w, c) \in \mathcal{P}_{\text{full}}\}. \quad (13)$$

A kernel representation of \mathcal{P}_c can be obtained by eliminating w from the kernel representation of $\mathcal{P}_{\text{full}}$. 132 133

Despite of the nice property given in the previous theorem, the canonical controller also has the property of being *maximally irregular*, in the following sense. 134 135 136

Theorem 4: (cf. [7]) Assume that the specification \mathcal{S} is implementable. The canonical controller \mathcal{C}_{can} is regular if and only if every controller that implements \mathcal{S} is regular. 137 138 139

In this paper, we want to show that if the specification \mathcal{S} is regularly implementable at all, then, although the canonical controller itself is maximally irregular, there exist regular controllers that are equivalent to it. By equivalent controllers, we mean controllers that allow the same set of c trajectories of the plant as the canonical controller does. The class of such controllers is defined as follows. 140 141 142 143 144 145

Definition 5: The class of regular controllers that are equivalent to the canonical controller is denoted by $\mathcal{C}_{\text{can}}^{\text{reg}}$, and is defined as 146 147

$$\mathcal{C}_{\text{can}}^{\text{reg}} := \{\mathcal{C} \mid \mathcal{C} \text{ is regular and } \mathcal{C} \cap \mathcal{P}_c = \mathcal{C}_{\text{can}} \cap \mathcal{P}_c\}. \quad (14)$$

The following theorem provides a necessary and sufficient conditions for the nonemptiness of the class $\mathcal{C}_{\text{can}}^{\text{reg}}$. This theorem is given without proof due to space limitation. The reader is referred to [8] and [14] for the proof, and to [15] and [16] for related results for nD behaviors. 148 149 150 151 152

Theorem 6: The class $\mathcal{C}_{\text{can}}^{\text{reg}}$ is nonempty if and only if the specification \mathcal{S} is regularly implementable. 153 154

In fact, regular implementability of the specification \mathcal{S} also implies that, for every regular controller that implements \mathcal{S} , there exists a superset of that controller in $\mathcal{C}_{\text{can}}^{\text{reg}}$ that implements \mathcal{S} . This is the content of the following theorem. 155 156 157 158

Theorem 7: [8], [14] Given a full plant behavior $\mathcal{P}_{\text{full}}$ and a regularly implementable specification \mathcal{S} . If \mathcal{C} is a regular controller that implements \mathcal{S} , then there exists a regular controller $\mathcal{C}' \in \mathcal{C}_{\text{can}}^{\text{reg}}$ that implements \mathcal{S} and $\mathcal{C} \subseteq \mathcal{C}'$. 159 160 161 162

Given the importance of the set $\mathcal{C}_{\text{can}}^{\text{reg}}$, in this paper, we present a parametrization of all controllers in $\mathcal{C}_{\text{can}}^{\text{reg}}$. Before we can obtain the parametrization, we need the following lemma. 163 164 165

166 *Lemma 8:* [8], [14] Let a plant \mathcal{P} be given as the kernel of a full row
 167 rank $R(d/dt)$ and a regular controller \mathcal{C} be given as the kernel of a full
 168 row rank $C(d/dt)$. Denote the full interconnection by $\mathcal{K} := \mathcal{P} \cap \mathcal{C}$.
 169 Let $\mathfrak{C}_{\mathcal{K}}$ denote the set of all controllers (not necessarily regular ones)
 170 that: 1) have at most as many outputs as \mathcal{C} and 2) also implement \mathcal{K}
 171 when interconnected with \mathcal{P} . A controller $\mathcal{C}' \in \mathfrak{C}_{\mathcal{K}}$ if and only if its
 172 kernel representation can be written as $VR + C$ for some matrix V .
 173 Moreover, every controller in $\mathcal{C}' \in \mathfrak{C}_{\mathcal{K}}$ has the properties that: 1) \mathcal{C}' is
 174 regular and 2) \mathcal{C}' has exactly as many outputs as \mathcal{C} . Notice that the
 175 number of outputs is $p(\mathcal{K}) - p(\mathcal{P})$.

176 If we pick any regular controller $\mathcal{C} \in \mathfrak{C}_{\text{can}}^{\text{reg}}$, Lemma 8 can be used
 177 to parametrize all other controllers in $\mathfrak{C}_{\text{can}}^{\text{reg}}$ based on a kernel repre-
 178 sentation of \mathcal{C} . This is one of the main results of this paper, which is
 179 summarized in the following theorem.¹

180 *Theorem 9:* [14] Let the control manifest behavior of the plant \mathcal{P}_c
 181 be the kernel of $P_c(d/dt)$ and a controller $\mathcal{C} \in \mathfrak{C}_{\text{can}}^{\text{reg}}$ be the kernel
 182 of $C(d/dt)$. Assume that both P_c and C are full row rank. A con-
 183 troller \mathcal{C}' is an element of $\mathfrak{C}_{\text{can}}^{\text{reg}}$ if and only if it is the kernel of
 184 $V(d/dt)P_c(d/dt) + C(d/dt)$ for some polynomial matrix $V(\xi)$.
 185 *Proof:* The full plant behavior can be represented by

$$\begin{bmatrix} \tilde{R} \left(\frac{d}{dt} \right) & \tilde{M} \left(\frac{d}{dt} \right) \\ 0 & P_c \left(\frac{d}{dt} \right) \end{bmatrix} \begin{bmatrix} w \\ c \end{bmatrix} = 0 \quad (15)$$

186 where \tilde{R} is full row rank. It follows that a controller \mathcal{C}' represented as
 187 the kernel of $\mathcal{C}'(d/dt)$ is regular if and only if

$$\text{rank} \begin{bmatrix} P_c \\ \mathcal{C}' \end{bmatrix} = \text{rank } P_c + \text{rank } \mathcal{C}'. \quad (16)$$

188 This is equivalent to saying that the interconnection of \mathcal{P}_c and \mathcal{C}' is
 189 regular. Therefore, we can apply Lemma 8 (by replacing \mathcal{K} with \mathcal{C}_{can}
 190 and \mathcal{P} with \mathcal{P}_c) and obtain the parametrization of all elements in $\mathfrak{C}_{\text{can}}^{\text{reg}}$.

191 IV. CONTROL WITH MINIMAL INTERACTION

192 Consider the following definition of irrelevant variables.

193 *Definition 10:* Let a behavior \mathfrak{B} be given by the kernel representation

$$R_1 \left(\frac{d}{dt} \right) w_1 + R_2 \left(\frac{d}{dt} \right) w_2 = 0. \quad (17)$$

194 The variables in w_1 are said to be *irrelevant* to \mathfrak{B} if \mathfrak{B} can be written
 195 as $\mathfrak{C}^{\infty}(R, R^{w_1}) \times \mathfrak{B}_2$, where \mathfrak{B}_2 is the behavior of w_2 .

196 Notice that w_1 being irrelevant to \mathfrak{B} in (17) is equivalent with
 197 $R_1 = 0$. The number of irrelevant variables in a behavior \mathfrak{B} is thus the
 198 number of zero columns in a kernel representation of it. For any system
 199 \mathfrak{B} , denote the number of its irrelevant variables by $i(\mathfrak{B})$. It can be
 200 proven that $i(\mathfrak{B})$ is independent of the choice of kernel representation
 201 of \mathfrak{B} . The problem of control with minimal interaction that we are
 202 addressing in this paper can be formulated as follows.

203 *Control with minimal interaction.* Given are the full plant behavior
 204 $\mathcal{P}_{\text{full}}$ (1) and specification \mathcal{S} . We assume that the specification \mathcal{S} is
 205 regularly implementable. Construct a regular controller \mathcal{C} such that: 1)
 206 \mathcal{C} implements \mathcal{S} and 2) if \mathcal{C}' is a regular controller that implements \mathcal{S} ,
 207 then $i(\mathcal{C}) \geq i(\mathcal{C}')$.

208 A controller that satisfies the aforementioned requirements is called
 209 a *controller with minimal interaction*. When some control variables

are irrelevant to the controller, we can realize the controller without
 using these variables. A controller with minimal interaction is thus a
 controller that uses the fewest number of variables in its realization.
 Notice that such a controller is generally not unique.

We use the parametrization of $\mathfrak{C}_{\text{can}}^{\text{reg}}$ that we derived in the previous
 section to solve the problem of control with minimal interaction. First,
 consider the following lemma.

217 *Lemma 11:* Let \mathfrak{B} be a behavior, whose variables include the variable
 218 w_1 . If w_1 is irrelevant to \mathfrak{B} , then it is also irrelevant to any $\mathfrak{B}' \supseteq \mathfrak{B}$.

Lemma 11 and Theorem 7 tell us that it is sufficient to search for a
 controller with minimal interaction in $\mathfrak{C}_{\text{can}}^{\text{reg}}$, instead of in the set of all
 regular controllers. This is an advantage, since we can parametrize all
 the controllers in $\mathfrak{C}_{\text{can}}^{\text{reg}}$, as shown in Theorem 9. To solve the problem
 of control with minimal interaction, we need to find an element of $\mathfrak{C}_{\text{can}}^{\text{reg}}$
 with the maximal number of zero columns. Generally, since there are
 finitely many columns, there is a maximal number of zero columns
 that can be attained. However, there is no guarantee that this number is
 attained by a unique controller. In fact, generally speaking, it is not.

The procedure to compute a regular controller that implements \mathcal{S}
 and has the maximal number of irrelevant variables can be summarized
 as follows.

Step 1) Construct the canonical controller \mathcal{C}_{can} for the problem.
 Since \mathcal{S} is regularly implementable, we know that the canon-
 ical controller implements \mathcal{S} .

Step 2) Construct a controller $\mathcal{C} \in \mathfrak{C}_{\text{can}}^{\text{reg}}$. Denote a kernel repre-
 sentation of \mathcal{C} and the control manifest behavior \mathcal{P}_c by $C(d/dt)$
 and $P(d/dt)$, respectively.

Step 3) The kernel representation of the controller with minimal
 interaction can be found by finding a matrix V such that
 $C + VP$ has the maximal number of zero columns.

The algebraic problem related to the third step has a combinatorial
 aspect in it, as we generally need to search for the answer by trying all
 possible subsets of the columns. This situation gives rise to a compu-
 tational challenge, namely to design an algorithm that can handle this
 combinatorial problem efficiently. We refer the reader to [18] for an
 algorithm that solves the combinatorial problem. The following lemma
 establishes an upper bound for the number of irrelevant variables that
 can be attained in the controller with minimal interaction.

248 *Lemma 12:* A controller with minimal interaction can have at most
 249 $c - p(\mathcal{C})$ irrelevant variables. Here, c denotes the number of all control
 250 variables (the number of components of \mathbf{c}) and $p(\mathcal{C})$ denotes the number
 251 of output variables in \mathcal{C} , which is the same for all regular controller that
 252 implements \mathcal{S} .

253 *Proof:* From the definition of regularity, we know that all regular
 254 controllers that implement \mathcal{S} have the same number of outputs, i.e.,
 255 $p(\mathcal{C})$. This is the number of rows in a minimal kernel representation
 256 of the controller. It is easily seen that the number of columns is c . If
 257 a regular controller has more than $c - p(\mathcal{C})$ irrelevant variables, then
 258 the nonzero entries of any kernel representation of it form a tall matrix,
 259 and thus cannot be minimal. ■

260 V. CONTROL PROBLEM WITH INPUT-OUTPUT PARTITION 261 CONSTRAINT

262 In some cases, it is physically necessary to require that in a con-
 263 troller, some of the plant control variables are free variables, for exam-
 264 ple, because these variables are sensor outputs. The control problem
 265 with input-output partitioning constraint for linear systems is formally
 266 defined as follows.

¹We refer the reader to [17] for related results.

267 *Control with input–output partition constraint.* Given a control prob-
268 lem, where the plant is

$$\mathcal{P} = \left\{ (w, c_1, c_2) \mid R \left(\frac{d}{dt} \right) w + P \left(\frac{d}{dt} \right) c_1 + Q \left(\frac{d}{dt} \right) c_2 = 0 \right\}. \quad (18)$$

269 The control variables are c_1 and c_2 , the to-be-controlled variable is
270 w . The desired specification is given as

$$\mathcal{S} = \left\{ w \mid S \left(\frac{d}{dt} \right) w = 0 \right\}. \quad (19)$$

271 Find a regular controller \mathcal{C} described as

$$\mathcal{C} = \left\{ (c_1, c_2) \mid C_1 \left(\frac{d}{dt} \right) c_1 + C_2 \left(\frac{d}{dt} \right) c_2 = 0 \right\} \quad (20)$$

272 such that \mathcal{C} implements \mathcal{S} and the variables in \mathcal{C} can be input–output
273 partitioned such that c_2 is free in \mathcal{C} , i.e., for any $c_2 \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^{c_2})$,
274 there exists a $c_1 \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^{c_1})$ such that $(c_1, c_2) \in \mathcal{C}$.

275 To solve the problem, we assume that the specification \mathcal{S} is regularly
276 implementable (otherwise the problem is clearly not solvable).

277 *Notation 13:* We denote the class of regular controllers that imple-
278 ments \mathcal{S} as $\mathcal{C}_S^{\text{reg}}$.

279 To find a solution to the problem, we need to use the following result.

280 *Lemma 14:* Given a system

$$\mathcal{C} = \left\{ (c_1, c_2) \mid C_1 \left(\frac{d}{dt} \right) c_1 + C_2 \left(\frac{d}{dt} \right) c_2 = 0 \right\}. \quad (21)$$

281 Without loss of generality, we assume that $[C_1 \ C_2]$ is full row rank.
282 The variable c_2 is free in \mathcal{C} if and only if C_1 is full row rank.

283 Using Lemma 14, we can reformulate the control problem as follows.

284 *Problem.* Find a controller $\mathcal{C} \in \mathcal{C}_S^{\text{reg}}$ in the form of

$$\mathcal{C} = \left\{ (c_1, c_2) \mid C_1 \left(\frac{d}{dt} \right) c_1 + C_2 \left(\frac{d}{dt} \right) c_2 = 0 \right\}$$

285 where C_1 is full row rank.

286 We shall use the following lemma to show that we can restrict our
287 attention to controllers in $\mathcal{C}_{\text{can}}^{\text{reg}}$ in solving the problem.

288 *Lemma 15:* Let X be a subset of $\mathcal{C}_S^{\text{reg}}$ such that for any $\mathcal{C} \in \mathcal{C}_S^{\text{reg}}$,
289 there exists a $\mathcal{C}' \in X$ such that $\mathcal{C} \subseteq \mathcal{C}'$. Then there exists a $\mathcal{C} \in \mathcal{C}_S^{\text{reg}}$ that
290 solves the control problem with input–output partitioning constraint if
291 and only if there exists a $\mathcal{C}' \in X$ that does so.

292 This lemma tells us that if we can construct a subset of $\mathcal{C}_S^{\text{reg}}$ with the
293 property of X , we do not need to search for the candidate controller in
294 the whole $\mathcal{C}_S^{\text{reg}}$. Rather, we can restrict our attention in X . Theorem 7
295 shows that $\mathcal{C}_{\text{can}}^{\text{reg}}$ has the desired property. Thus, we shall try to construct
296 the desired controller in $\mathcal{C}_{\text{can}}^{\text{reg}}$, which we can parametrize according to
297 Theorem 9.

298 A solution to the control problem can be found by executing the
299 following steps.

300 Step 1) Construct the canonical controller \mathcal{C}_{can} for the problem.
301 Since \mathcal{S} is regularly implementable, we know that the canon-
302 ical controller implements \mathcal{S} .

303 Step 2) Construct a controller $\mathcal{C} \in \mathcal{C}_{\text{can}}^{\text{reg}}$. The proof of Theorem 6
304 contains information on how to construct \mathcal{C} from a regular
305 controller. Denote the kernel representation of \mathcal{C} and the
306 control manifest behavior \mathcal{P}_c , respectively, by

$$\mathcal{C} = \left\{ (c_1, c_2) \mid C_1 \left(\frac{d}{dt} \right) c_1 + C_2 \left(\frac{d}{dt} \right) c_2 = 0 \right\} \quad (22a)$$

$$\mathcal{P}_c = \left\{ (c_1, c_2) \mid P_1 \left(\frac{d}{dt} \right) c_1 + P_2 \left(\frac{d}{dt} \right) c_2 = 0 \right\}. \quad (22b)$$

Step 3) Following Theorem 9, any controller \mathcal{C}' in $\mathcal{C}_{\text{can}}^{\text{reg}}$ can be rep-
resented by

$$\mathcal{C}' = \left\{ (c_1, c_2) \mid (C_1 + VP_1) \left(\frac{d}{dt} \right) c_1 + (C_2 + VP_2) \left(\frac{d}{dt} \right) c_2 = 0 \right\}. \quad (23)$$

The kernel representation of a controller in $\mathcal{C}_{\text{can}}^{\text{reg}}$ that satisfies
the input–output partitioning constraint can be found by
finding a matrix V such that $C_1 + VP_1$ is full row rank.

A necessary and sufficient condition for the existence of such a
matrix V is given in the following lemma.

Lemma 16: Given polynomial matrices $C \in \mathbb{R}^{c \times q}[\xi]$ and $P \in \mathbb{R}^{p \times q}[\xi]$. There exists a polynomial matrix $V \in \mathbb{R}^{c \times p}[\xi]$ such that
 $C + VP$ is full row rank if and only if

$$\text{rank} \begin{bmatrix} P \\ C \end{bmatrix} \geq c.$$

We refer the reader to [18] for a proof of this lemma.

To conclude, the following is the algorithm to solve the control
problem with input–output partitioning constraint.

Algorithm 17: The following steps provide a solution to the problem
if and only if it is solvable.

- 1) Verify if the specification \mathcal{S} is regularly achievable. If so, go to
step 2, otherwise the problem is not solvable.
- 2) Construct the canonical controller for this problem, denote it by
 \mathcal{C}_{can} .
- 3) Construct a regular controller $\mathcal{C} \in \mathcal{C}_{\text{can}}^{\text{reg}}$. Theorem 6 guarantees
that this can be done. The controller \mathcal{C} and the control manifest
behavior \mathcal{P}_c can be represented in the form shown in (22).
- 4) Verify if $\text{rank} \begin{bmatrix} M_1 \\ P_1 \end{bmatrix} \geq p(\mathcal{C})$, where $p(\mathcal{C})$ denotes the number of
output variables of \mathcal{C} . If this condition is satisfied, go to step 5,
otherwise the problem is not solvable.
- 5) Compute a V such that $C_1 + VP_1$ is full row rank. The existence
of such V is guaranteed by Lemma 16. A controller that solves
the control problem is given by

$$\mathcal{C}' = \left\{ (c_1, c_2) \mid (C_1 + VP_1) \left(\frac{d}{dt} \right) c_1 + (C_2 + VP_2) \left(\frac{d}{dt} \right) c_2 = 0 \right\}. \quad (24)$$

VI. CONCLUDING REMARKS

The main result of the paper is a parametrization of all regular
controllers that are equivalent to the canonical controller $\mathcal{C}_{\text{can}}^{\text{reg}}$. This
class of controllers has the following two nice properties.

- 1) All its members are regular controllers.
- 2) It acts as an upperbound to other regular controllers. This means,
any regular controller is a subset of an element of $\mathcal{C}_{\text{can}}^{\text{reg}}$.

The special properties of the class $\mathcal{C}_{\text{can}}^{\text{reg}}$ and its parametrization are
used to solve two control problems in the behavioral framework. The
first control problem is related to designing a regular controller that
uses as few control variables as possible. The second problem is about

347 designing a regular controller that satisfies a predefined input–output
348 partitioning.

349 The use of the parametrization of $\mathcal{C}_{\text{can}}^{\text{reg}}$ is not necessarily limited
350 to the aforementioned problems. An interesting problem is, for exam-
351 ple, to use the parametrization to construct a regular controller with a
352 MacMillan degree as small as possible [2]. Such a result can potentially
353 lead to the solution to the long standing problem of regular feedback
354 implementability [19].

355

REFERENCES

- 356 [1] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems*
357 *Theory: A Behavioral Approach*. New York: Springer, 1998.
358 [2] J. C. Willems, “On interconnections, control and feedback,” *IEEE Trans.*
359 *Autom. Control*, vol. 42, no. 3, pp. 326–339, Mar. 1997.
360 [3] M. N. Belur and H. L. Trentelman, “Stabilization, pole placement and
361 regular implementability,” *IEEE Trans. Autom. Control*, vol. 47, no. 5,
362 pp. 735–744, May 2002.
363 [4] M. N. Belur, “Control in a behavioral context,” Ph.D. dissertation, Univ.
364 Groningen, Groningen, The Netherlands, Jun. 2003.
365 [5] A. J. van der Schaft and A. A. Julius, “Achievable behavior by com-
366 position,” in *Proc. 41st IEEE Conf. Decision Control*, Las Vegas, 2002,
367 pp. 7–12.
368 [6] A. J. van der Schaft, “Achievable behavior of general systems,” *Syst.*
369 *Control Lett.*, vol. 49, pp. 141–149, 2003.
370 [7] A. A. Julius, J. C. Willems, M. N. Belur, and H. L. Trentelman, “The
371 canonical controllers and regular interconnection,” *Syst. Control Lett.*,
372 vol. 54, pp. 787–797, 2005.
373 [8] A. A. Julius, J. W. Polderman, and A. J. van der Schaft, “Controller with
374 minimal interaction,” presented at the IFAC World Congr., Prague, Czech
375 Republic, 2005.
376 [9] J. C. Willems, “Behaviors, latent variables, and interconnections,” *Syst.*
377 *Control Inf. (Japan)*, vol. 43, pp. 453–464, 1999.

- [10] J. C. Willems and H. L. Trentelman, “Synthesis of dissipative systems
378 using quadratic differential forms—Part I,” *IEEE Trans. Autom. Control*,
379 vol. 47, no. 1, pp. 53–69, Jan. 2002. 380
[11] J. W. Polderman, “Sequential continuous time adaptive control: A be-
381 havioral approach,” in *Proc. 39th IEEE Conf. Decision Control*, Sydney,
382 2000, pp. 2484–2487. 383
[12] A. A. Julius and A. J. van der Schaft, “Compatibility of behavior inter-
384 connections,” presented at the Eur. Control Conf., Cambridge, U.K., Sep.
385 2003. 386
[13] J. C. Willems, M. N. Belur, A. A. Julius, and H. L. Trentelman, “The
387 canonical controller and its regularity,” in *Proc. IEEE Conf. Decision*
388 *Control*, Hawaii, Dec. 2003, pp. 1639–1644. 389
[14] A. A. Julius, “On interconnection and equivalence of continuous and dis-
390 crete systems: A behavioral perspective,” Ph.D. dissertation, Univ. Twente,
391 Enschede, The Netherlands, Feb. 2005. 392
[15] P. Rocha, “Canonical controllers and regular implementation of nD behav-
393 iors,” presented at the 16th IFAC World Congr., Prague, Czech Republic,
394 2005. 395
[16] H. L. Trentelman and D. N. Avelli, “On the regular implementability of
396 nd systems,” *Syst. Control Lett.*, vol. 56, no. 4, pp. 265–271, 2007. 397
[17] C. Praagman, H. L. Trentelman, and R. Z. Yoe, “On the parametrization of
398 all regularly implementing and stabilizing controllers,” *SIAM J. Control*
399 *Optim.*, vol. 45, no. 6, pp. 2035–2053, 2007. 400
[18] A. A. Julius, J. W. Polderman, and A. J. van der Schaft, “Parametrization of
401 the regular equivalences of the canonical controller and its applications,”
402 arXiv article nr. 0708.1309, Aug. 2007. 403
[19] H. L. Trentelman, “Regular feedback implementability for linear differ-
404 ential behaviors,” in *Unsolved Problems in Mathematical Systems and*
405 *Control Theory*. Princeton, NJ: Princeton Univ. Press, 2004, pp. 44–48. 406

408 Q1. Author: Please update Ref. [18].

IEEE
Proof