

A behavioral framework for compositionality : linear systems, discrete event systems and hybrid systems.

A. A. Julius and A. J. van der Schaft

Dept. Applied Mathematics

University of Twente

PO BOX 217, Enschede 7500AE

The Netherlands

Email : a.a.julius@math.utwente.nl

May 15, 2004

Abstract

In this paper we formulate a general framework based on the behavioral approach to dynamical systems, in which various issues regarding interconnection of systems can be addressed. The main part of the framework is that interconnections or compositions of systems can be modelled with interconnection of behaviors and generalized projection operations. Control problems such as supervisory control problem or feedback control problem can be expressed in terms of behavioral interconnection, and therefore can be put into this framework. In the paper we discuss some variants of control problems and provide solutions to them.

1 Introduction

The concept of interconnection has been used extensively to model complex systems. In this paper, we attempt to formulate a general framework based on the behavioral approach to dynamical systems, in which various issues regarding interconnection of systems can be addressed. Ideally, the framework should be broad enough so that a wide variety of system classes, including hybrid systems can be handled. For the hybrid systems community, the benefit of having such a framework is that analysis of hybrid systems can be done in a systematic manner, using concepts that have counterparts in other classes of systems, for example, discrete event systems and linear time invariant systems. The framework can also help in translating ideas and concepts of analysis for certain classes to others.

Having stated all the above, we should also remark that the idea of using behavioral approach as a framework for working with many classes of system is not at all new. In his seminal paper [1], which is one of the first literatures in the behavioral systems theory, Willems hinted on extending the discussion (which was mainly about linear systems) to discrete event systems. The behavioral approach to many classes of dynamical systems has been presented in a host of literatures, for example in linear systems [2, 3, 4], discrete event systems [5], and hybrid systems [6, 7, 8]. A preliminary step for bringing forth a general theory for these classes has been taken for example in [9, 10], where control problem in a general behavioral

setting is discussed, and [11], where compatibility for general behavior interconnections is discussed.

This paper is organized as follows. The following section contains some mathematical preliminaries, in which the framework is set for subsequent discussion. Section 3 explains about control problems from the point of view of behavior interconnections and discusses some variants of control problems in the behavioral framework as well as some examples. The final section contains some concluding remarks.

2 Mathematical preliminaries

The following definition of *dynamical systems* is adopted from [3].

Definition 1 *A dynamical system Σ is defined as a triple $(\mathbb{T}, \mathbb{W}, \mathfrak{B})$, where \mathbb{T} is called the **time axis**, \mathbb{W} the **signal space**, and \mathfrak{B} a set of functions (not necessarily total¹) $w : \mathbb{T} \rightarrow \mathbb{W}$, is called the **behavior** of the system. The time axis \mathbb{T} is a set with a total ordering $<$. We can regard \mathbb{W} and \mathbb{T} as the type of the behavior \mathfrak{B} .*

2.1 Projection

When modelling general interconnected behaviors, the concept of projection plays a central role. Let us first formally define what is meant by projection in this context.

Definition 2 *A projection $\pi : \mathfrak{B}_1 \rightarrow \mathfrak{B}_2$ is a total function mapping a behavior \mathfrak{B}_1 of type $(\mathbb{T}_1, \mathbb{W}_1)$ to another behavior \mathfrak{B}_2 of type $(\mathbb{T}_2, \mathbb{W}_2)$. Hence, the types of both behaviors need not be the same. Although projections are defined to be mapping between behaviors, in this paper we also naturally extend the use of projections to denote set-valued maps between behaviors, that is*

$$\pi(X) := \{w \in \mathfrak{B}_2 \mid \exists x \in X, \text{ such that } \pi(x) = w\}, \quad \forall X \subset \mathfrak{B}_1. \quad (1)$$

Definition 3 *Given a projection $\pi : \mathfrak{B}_1 \rightarrow \mathfrak{B}_2$, the set-valued inverse of π , denoted as π^{-1} is defined for any $X \subset \mathfrak{B}_2$ as*

$$\pi^{-1}(X) := \{w \in \mathfrak{B}_1 \mid \pi(w) \in X\}. \quad (2)$$

In the following, we shall consider two operations on projections that give a new projection.

(composition) It is easy to see that a composition of projections is again a projection, and that for two projections π and γ such that $\pi \circ \gamma$ is defined,

$$(\pi \circ \gamma)^{-1} = \gamma^{-1} \circ \pi^{-1}. \quad (3)$$

(product) Given two projections $\pi : \mathfrak{B}_1 \rightarrow \mathfrak{B}_2$ and $\gamma : \mathfrak{B}_1 \rightarrow \mathfrak{B}_3$. If \mathfrak{B}_2 and \mathfrak{B}_3 have equal time axis, we define $\pi \times \gamma$ as

$$(\pi \times \gamma)(w) := (\pi(w), \gamma(w)), \quad \forall w \in \mathfrak{B}_1. \quad (4)$$

¹by total function we mean $\text{dom}(w) = \mathbb{T}$.

Here $(\pi \times \gamma) : \mathfrak{B}_1 \rightarrow \mathfrak{B}_4$, where \mathfrak{B}_4 is of type $(\mathbb{T}, \mathbb{W}_2 \times \mathbb{W}_3)$, where \mathbb{T} is the time axis shared by \mathfrak{B}_2 and \mathfrak{B}_3 , and \mathbb{W}_2 and \mathbb{W}_3 are their signal spaces respectively. The inverse of the product is defined as follows.

$$(\pi \times \gamma)^{-1}(a, b) := \pi^{-1}(a) \cap \gamma^{-1}(b), \quad \forall a \in \mathfrak{B}_2, b \in \mathfrak{B}_3. \quad (5)$$

Example 4 Take a linear system $\Sigma := (\mathbb{R}, \mathbb{R}^n, \mathfrak{B})$, where \mathfrak{B} is a set of n -tuple smooth functions $w := (w_1, w_2, \dots, w_n)$ such that

$$R \left(\frac{d}{dt} \right) w = 0.$$

Here $R(\xi)$ is a polynomial matrix of size $m \times n$ with ξ as its indeterminate. An example of typical projection operators of this type of system is $\pi_k : \mathfrak{B} \rightarrow \mathfrak{B}'$. The behavior \mathfrak{B}' is of the type $(\mathbb{R}, \mathbb{R}^k)$, $k < n$. For any $w := (w_1, w_2, \dots, w_n) \in \mathfrak{B}$, we define

$$\pi_k w := w' := (w_1, w_2, \dots, w_k). \quad (6)$$

As the result of the elimination theorem [3], $\pi_k \mathfrak{B}$ can be written as the kernel of another differential operator $R' \left(\frac{d}{dt} \right)$.

Example 5 Take a prefix closed language L consisting of strings of symbols taken from the event set E . The strings can be thought of as the sequences of events generated by an automaton². The language L can be regarded as a behavior of type (\mathbb{Z}_+, E) . An example of typical projection operations of this type of system is $\pi_F : L \rightarrow L'$. The behavior L' is of the type (\mathbb{Z}_+, F) , $F \subset E$. For any string $s \in L$, the string $\pi_F s$ is formed by removing all events in s which are not contained in F .

Example 6 Consider a hybrid automaton [8] \mathcal{A} , with the set of external variables W and internal variables X . The set of actions are also divided into external ones E and internal ones H . Consider the collection of **execution fragments** of the automaton. This collection can be regarded as a behavior with type $(\mathbb{R} \times \mathbb{Z}_+, \mathbb{R}^v \cup A)$, where $V := W \cup X$, and v is its cardinality, and $A := E \cup H$. See Figure 1 for an illustration. The **traces** of the automaton are obtained by projecting the continuous trajectories of the execution fragments to the set of external variables (as in Example 4) and projecting the strings of events of the execution fragments to the set of external actions (as in Example 5). The collection of traces is therefore a behavior of type $(\mathbb{R} \times \mathbb{Z}_+, \mathbb{R}^w \cup E)$, where w is the cardinality of W .

Example 7 Consider a continuous time linear behavior in input-state-output representation

$$\dot{x} = Ax + Bu, \quad (7a)$$

$$y = Cx. \quad (7b)$$

Denote the dimensions of the state, input and output as \mathbf{n} , \mathbf{m} , and \mathbf{q} respectively. A basic scheme of discrete-time feedback control of such systems is depicted in Figure 2. The idea is to

²We refer the reader to [12] for more detailed expositions.

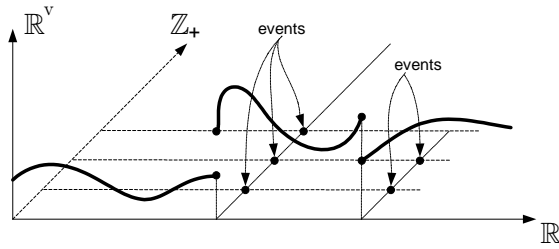


Figure 1: Illustration of an execution fragment of a hybrid automaton.

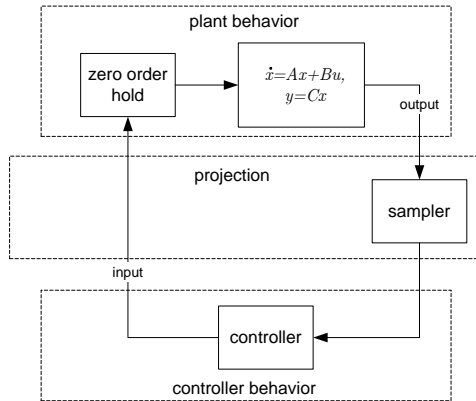


Figure 2: Block diagram of the discrete-time control of a continuous plant.

sample the output y and use a discrete-time behavior as a controller, and then feed the discrete-time input obtained from the controller to the plant. This input is then "made continuous" by using a zero order hold. The plant behavior \mathcal{P} is the collection of all (x, u, y) trajectories satisfying (7), such that u is piecewise constant between the sampling time instants. The type of this behavior is $(\mathbb{R}, \mathbb{R}^{n+m+q})$. For simplicity, assume that the sampling period is 1 second. The projection associated to the sampling is then defined as follows.

$$\pi(x, u, y) := (\tilde{u}, \tilde{y}), \quad \forall (x, u, y) \in \mathcal{P}, \quad (8)$$

where \tilde{u} and \tilde{y} are discrete time signals such that for all $k \in \mathbb{Z}_+$,

$$\begin{aligned} \tilde{u}(k) &= u(k), \\ \tilde{y}(k) &= y(k). \end{aligned}$$

Clearly this projection produces a behavior of type $(\mathbb{Z}, \mathbb{R}^{m+q})$.

The examples above illustrate how the notion of projection, which has different meaning on different kinds of systems, can be brought into a more general setting. The fact that projection plays a central role in the behavioral approach to systems theory will be more pronounced when we discuss about interconnection.

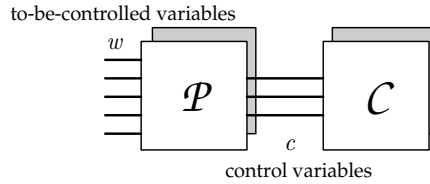


Figure 3: The setup of usual linear control interconnection.

2.2 Interconnection/composition

In the following we shall discuss the notion of *interconnection* or *composition*³.

Definition 8 Given two behaviors of the *same type* \mathfrak{B}_1 and \mathfrak{B}_2 , the *interconnection* or *composition* of them, denoted as $\mathfrak{B}_1 \parallel \mathfrak{B}_2$ is defined to be another behavior of the same type given by $\mathfrak{B}_1 \cap \mathfrak{B}_2$.

This definition of interconnection coincides with the notion of interconnection of linear behaviors [2] and automata [12]. Given two linear behaviors $\mathfrak{B}_i := \ker R_i \left(\frac{d}{dt} \right)$, $i = 1, 2$. The composition of these two behaviors is given by

$$\begin{aligned} \mathfrak{B} &:= \mathfrak{B}_1 \parallel \mathfrak{B}_2, \\ &= \ker R_1 \left(\frac{d}{dt} \right) \cap \ker R_2 \left(\frac{d}{dt} \right), \\ &= \ker \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \left(\frac{d}{dt} \right). \end{aligned} \quad (9)$$

Similarly, given two automata sharing the same alphabet A , generating the languages L_1 and L_2 , the language of the composed automaton is $L_1 \cap L_2$.

In most cases, it is desirable to define interconnection of behaviors of different types. For example, in linear systems not all subsystems share the same set of variables. Similarly, it is often desirable to assume that not all symbols in the alphabet are used in synchronization of automata. This kind of interconnections are often called *partial interconnections* [10]. This type of interconnections is formalized using projections, since projections can change the type of behaviors.

Example 9 See Figure 3. The usual setup of control in linear behaviors is that the plant \mathcal{P} is assumed to have two sets of variables w (to-be-controlled variables) and c (control variables). They are not necessarily disjoint. See for example [4] and [10]. The controller \mathcal{C} has c as its set of variables. We define π_c and π_w to be the projections that work on \mathcal{P} by projecting the trajectories into its c and w component respectively, as in Example 4. The controlled behavior of the plant (in terms of the w variables only) is then expressed as $\pi_w \pi_c^{-1} (\pi_c \mathcal{P} \parallel \mathcal{C})$.

Example 10 In discrete event systems, the setup is similar to that of the previous example, where instead of partitioning the variables, we partition the alphabet of the plant. The projection is then defined according to Example 5.

³The terms *interconnection* and *composition* are used interchangeably in this text.

The whole discussion about interconnection and projection so far can be summed up as the following. We use behaviors to represent system entities. These system entities can interact with its environment, or to be precise other system entities in the environment. Projection plays the role of defining how these entities interact, since one system can interact with different other entities in many different ways.

2.3 Observability

Consider a behavior \mathfrak{B}_1 together with a projection $\pi : \mathfrak{B}_1 \rightarrow \mathfrak{B}_2$. Denote I as the identity relation on \mathfrak{B}_1 and J be the equivalence induced by π , i.e.

$$J := \{(v, w) \in \mathfrak{B}_1 \times \mathfrak{B}_1 \mid \pi(v) = \pi(w)\}. \quad (10)$$

Clearly we have that

$$I \subseteq J. \quad (11)$$

When $J = I$, we say that the behavior \mathfrak{B}_1 is *observable* from the projection π . When the inclusion in (11) is strict, we say that \mathfrak{B}_1 is *not observable* from the projection π . Otherwise stated, the behavior \mathfrak{B}_1 is observable from the projection π if and only if $\pi \circ \pi^{-1}$ is the identity map of \mathfrak{B}_1 . This statement is not precise, since the codomain of π^{-1} is the power set of \mathfrak{B}_1 instead of \mathfrak{B}_1 itself. What is meant in fact, is that for any $w \in \mathfrak{B}_1$, $(\pi \circ \pi^{-1})(w)$ returns the singleton $\{w\}$. If \mathfrak{B}_1 is not observable from the projection π , the range of $\pi \circ \pi^{-1}$ is a family of subsets of \mathfrak{B}_1 , where each subset contains trajectories indistinguishable by the projection π .

Similarly, we extend the definition to observability of a projection γ from another projection π . Assume that both γ and π act on the same behavior \mathfrak{B}_1 . Let J be the equivalence relation induced by π , as defined in (10), and K be that of γ . We say that γ is observable from π if $J \subseteq K$. Equivalently, $\gamma^{-1} \circ \pi \circ \pi^{-1} \circ \gamma$ is the identity map on the range of γ .

The whole discussion on observability so far is captured in Figure 4. In the illustration, we see that \mathfrak{B}_1 and γ are observable from π , but π is not observable from γ .

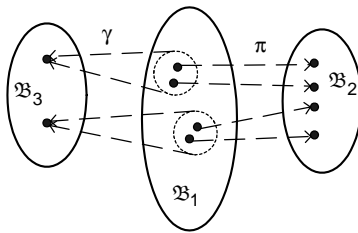


Figure 4: Illustration for observability.

Based on observability, we can define a partial ordering on the set of projections acting on a behavior. Let π and γ be two projections acting on \mathfrak{B}_1 . Let J and K be the equivalence relations induced by the two projections respectively. We define

$$\pi \lesssim \gamma :\Leftrightarrow J \supseteq K. \quad (12)$$

Two projections π and γ are equivalent, denoted by $\pi \approx \gamma$, iff $\pi \lesssim \gamma \lesssim \pi$. Equivalent projections are also called *isomorphic*. The reason behind the naming is the existence of

one-to-one mappings between the range of the projections. The following lemma states the properties of \lesssim with respect to the product and composition of two projections.

Lemma 11 *Let π and γ be any two projections acting on the same behavior such that $\pi \times \gamma$ is defined, and ϕ be such that $\pi \circ \phi$ is defined. The following relations hold*

$$\pi \times \pi \approx \pi, \quad (13a)$$

$$\pi \lesssim \pi \times \gamma \gtrsim \gamma, \quad (13b)$$

$$(\pi \lesssim \gamma) \Leftrightarrow \pi \times \gamma \approx \gamma, \quad (13c)$$

$$\pi \circ \phi \lesssim \pi. \quad (13d)$$

The whole discussion about observability among projections above can be summarized in the following lemma.

Lemma 12 *Given two projections γ and π acting on the same behavior \mathfrak{B} . The projection γ is observable from π if and only if $\gamma \lesssim \pi$.*

Example 13 *Given a behavior of a linear system \mathfrak{B} , with two set of variables w and z . The variables in w are said to be observable from z if for any (w_1, z_1) and (w_2, z_2) in \mathfrak{B} , the following implication holds [3]*

$$(z_1 = z_2) \Rightarrow (w_1 = w_2).$$

Because of the linearity of the behavior, this is equivalent to

$$(w, 0) \in \mathfrak{B} \Rightarrow (w = 0).$$

For systems in state-space representation, the observability of the states from the input-output variables corresponds to the well known Kalman rank condition and Hautus test.

For prefix closed languages, where projections are characterized by subsets of the alphabet, as explained in Example 5, the following theorem describes the relation between projections.

Theorem 14 *Given L a prefix closed language, with E as its alphabet. Assume that every symbol in E appears at least once⁴ in L . Denote the projections with respect to the subsets of symbols E_1 and E_2 as π and γ respectively. The following relation holds.*

$$(E_1 \subseteq E_2) \Leftrightarrow (\pi \lesssim \gamma).$$

Proof. (\Rightarrow) Define π' to be the projection with respect to the set of labels E_1 , acting on the codomain of γ . Clearly, we have that $\pi = \gamma \circ \pi'$. By Lemma 11, this implies $\pi \lesssim \gamma$.

(\Leftarrow) Suppose that $E_1 \not\subseteq E_2$. Let e be an element of E_1 , which is not in E_2 . Since L is prefix closed and e appears at least once in the language, there is a string (possibly empty) $s \in L$, such that $se \in L$. The pair (s, se) is in the equivalence relation induced by γ but not in that of π . By (12), it follows that $\pi \not\lesssim \gamma$. ■

An immediate consequence of Lemma 12 and Theorem 14 is that a prefix closed language L is observable from a projection if and only if the projection contains all the symbols appearing in L .

⁴This can be done without any loss of generality, since symbols that never appear can be discarded from the alphabet in the first place.

3 Control problems in behavioral context

One of the main themes that appear in the literature of behavioral systems theory, is about control. See for example [2, 4, 5, 6, 10, 13].

There are many variants of control problems, which share the same salient feature. They are all about finding a behavior (called the *controller*), which when composed with the plant behavior in a certain manner yields some desired properties, usually given in terms of another behavior (called the *specification*). In this section, we discuss some variants of the control problem and their conditions of solvability.

3.1 Full interconnection control problems

Full interconnection control problems are arguably the simplest variant. The behaviors involved are of the same type, so that no projection is necessary. The problem is typically expressed as the following. Given the plant \mathcal{P} and the specification \mathcal{S} , find a controller \mathcal{C} such that

$$\mathcal{P} \parallel \mathcal{C} = \mathcal{S}. \quad (14)$$

Notice that (14) implicitly suggests that all the behaviors involved are of the same type. Examples of control problems of this type are state feedback control problem (where the controller can use all the plant variables) and supervisory control of discrete event systems where all events are observable and controllable [12].

A controller behavior is said to *achieve* the desired specification \mathcal{S} , if it satisfies (14). The specification \mathcal{S} is said to be *achievable*, if there exists a controller that achieves it.

From Definition 8, it is clear that \mathcal{S} is achievable if and only if $\mathcal{S} \subset \mathcal{P}$. In fact, if \mathcal{S} is achievable, it can be seen easily that $\mathcal{C} := \mathcal{S}$ achieves \mathcal{S} .

3.2 Partial interconnection control problems

This variant of control problems involves behaviors of different types. Figure 3 depicts an example of such problems. Typically, the plant \mathcal{P} is a given behavior of type $(\mathbb{T}_P, \mathbb{W}_P)$, and the desired specification \mathcal{S} is given behavior of type $(\mathbb{T}_S, \mathbb{W}_S)$. The candidate controllers can be of yet another different type, say $(\mathbb{T}_C, \mathbb{W}_C)$. A controller \mathcal{C} of this type is said to achieve the specification \mathcal{S} (and hence solve the problem) if it satisfies

$$\pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}) = \mathcal{S}. \quad (15)$$

The projections π_c and π_s are given as part of problem. They map \mathcal{P} to behaviors of type equal to that of \mathcal{C} and \mathcal{S} respectively. Notice that if these two projections are the identity map, then the problem is essentially reduced into a full interconnection problem discussed in the previous subsection.

Control problems of this type have been discussed in somewhat different representation in, for example [9, 10]. There, the conditions for solving the problem as well as a candidate solution, called the *canonical controller*, were presented.

Define the *first canonical controller* as the following [14],

$$\mathcal{C}'_{\text{can}} := \{c \in \pi_c \mathcal{P} \mid \pi_s \pi_c^{-1} c \subset \mathcal{S}\}. \quad (16)$$

The first canonical controller possesses an important property, stated in the following proposition.

Proposition 15 *The controller \mathcal{C}'_{can} achieves the maximal achievable behavior contained in \mathcal{S} .*

Proof. From (16) we can readily conclude that

$$\pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}'_{can}) \subset \mathcal{S}. \quad (17)$$

To show that \mathcal{C}'_{can} achieves the maximal achievable behavior in \mathcal{S} , consider any other controller \mathcal{C}' such that

$$\pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}') \subset \mathcal{S}. \quad (18)$$

We shall show that $(\pi_c \mathcal{P} \parallel \mathcal{C}') \subset (\pi_c \mathcal{P} \parallel \mathcal{C}'_{can})$. Take any $c \in (\pi_c \mathcal{P} \parallel \mathcal{C}')$. Because of (18), we know that $\pi_s \pi_c^{-1}c \in \mathcal{S}$. Therefore, by the construction in (16), c is also contained in $(\pi_c \mathcal{P} \parallel \mathcal{C}'_{can})$. It follows that the behavior achieved by \mathcal{C}' is contained in that of \mathcal{C}'_{can} . ■

Theorem 16 *The specification \mathcal{S} in (15) is achievable, if and only if \mathcal{C}'_{can} achieves \mathcal{S} .*

Proof. We only have to prove the only if part, since the converse is obvious. From Proposition 15 we know that \mathcal{C}'_{can} achieves the maximal achievable behavior contained in \mathcal{S} . This means that if \mathcal{S} itself is achievable then \mathcal{C}'_{can} achieves \mathcal{S} . ■

The plant \mathcal{P} is said to satisfy the *homogeneity property* [9] with respect to the projections π_s and π_c , if for any (s_1, c_1) and (s_2, c_1) in $\pi_s \mathcal{P} \times \pi_c \mathcal{P}$,

$$(s_1, c_2) \in \pi_s \mathcal{P} \times \pi_c \mathcal{P} \Rightarrow (s_2, c_2) \in \pi_s \mathcal{P} \times \pi_c \mathcal{P}, \quad \forall c_2 \in \pi_c \mathcal{P}. \quad (19)$$

Remark 17 *In terms of observability, the homogeneity property can be also expressed as follows. The homogeneity property is satisfied if and only if $\pi_s \pi_c^{-1} \pi_c \approx \pi_c \pi_s^{-1} \pi_s$. Both projections are defined to be acting on \mathcal{P} .*

The conditions for solvability of the control problem is given in the following theorem, which was proved in [9].

Theorem 18 *For a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , the specification \mathcal{S} as in (15) is achievable if and only if*

- (i) $\mathcal{S} \subset \pi_s \mathcal{P}$, and
- (ii) $\mathcal{S} = \pi_s \pi_c^{-1} \pi_c \pi_s^{-1} \mathcal{S}$.

For special cases, where the homogeneity property is satisfied, we can construct the *second canonical controller* as follows⁵.

$$\mathcal{C}''_{can} = \pi_c \pi_s^{-1}(\pi_s \mathcal{P} \parallel \mathcal{S}). \quad (20)$$

The second canonical controller has the following special property.

Proposition 19 *For a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , the second canonical controller \mathcal{C}''_{can} achieves the minimal achievable behavior containing $(\mathcal{S} \parallel \pi_s \mathcal{P})$.*

⁵In [9, 10] the second canonical controller is called canonical controller. The first canonical controller is introduced later in [14].

Proof. Denote $\mathcal{K}_{\text{can}}'' := \pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}_{\text{can}}'')$. Since $\mathcal{C}_{\text{can}}'' \subset \pi_c \mathcal{P}$,

$$\begin{aligned} \mathcal{K}_{\text{can}}'' &= \pi_s \pi_c^{-1}(\mathcal{C}_{\text{can}}''), \\ &= \pi_s \pi_c^{-1} \pi_c \pi_s^{-1}(\pi_s \mathcal{P} \parallel \mathcal{S}), \\ &\supset (\pi_s \mathcal{P} \parallel \mathcal{S}). \end{aligned} \tag{21}$$

Take any other controller \mathcal{C}' such that $(\pi_s \mathcal{P} \parallel \mathcal{S}) \subset \pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}')$. Denote $\mathcal{K}' := \pi_s \pi_c^{-1}(\pi_c \mathcal{P} \parallel \mathcal{C}')$. Take any $s \in \mathcal{K}_{\text{can}}''$. We shall prove that $s \in \mathcal{K}'$. If $s \in (\pi_s \mathcal{P} \parallel \mathcal{S})$ then $s \in \mathcal{K}'$ since $(\pi_s \mathcal{P} \parallel \mathcal{S}) \subset \mathcal{K}'$. If $s \notin (\pi_s \mathcal{P} \parallel \mathcal{S})$, then there exist w and w' in \mathcal{P} such that $\pi_s w = s$, $\pi_c w =: c \in \mathcal{C}_{\text{can}}''$, $\pi_c w' = c$, and $\pi_s w' =: s' \in (\pi_s \mathcal{P} \parallel \mathcal{S})$. We are going to show that $s \notin \mathcal{K}'$ is a contradiction. Suppose that it is true, then $\pi_c \pi_s^{-1} s \parallel \mathcal{C}' = \emptyset$. But because of the homogeneity property, $\pi_c \pi_s^{-1} s = \pi_c \pi_s^{-1} s'$. Hence $\pi_c \pi_s^{-1} s' \parallel \mathcal{C}' = \emptyset$ and $s' \notin \mathcal{K}'$, which is a contradiction since $s' \in (\pi_s \mathcal{P} \parallel \mathcal{S})$. ■

The following corollary, follows from Theorem 18 and Proposition 19.

Corollary 20 *For a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , \mathcal{S} is achievable if and only if $\mathcal{C}_{\text{can}}''$ achieves it.*

Stronger than this corollary, we can state the following proposition. A proof is not given due to space limitation.

Proposition 21 *For a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , if \mathcal{S} is achievable then $\mathcal{C}_{\text{can}}' = \mathcal{C}_{\text{can}}''$.*

An example of control problems of this type is supervisory control of discrete event systems, where the set of controllable events coincides with the observable ones⁶.

Corollary 22 *Let L be a prefix closed language, with alphabet E . Let $Z \subset E$ be the set of observable and controllable events. There exists a supervisor language C such that the specification \mathcal{S} is achieved, i.e. $\pi_z^{-1}(\pi_z L \parallel C) = \mathcal{S}$, where π_z is the natural projection associated with the set of events Z , if and only if*

- (i) $\mathcal{S} \subset L$, and
- (ii) $\mathcal{S} = \pi_z^{-1} \pi_z \mathcal{S}$.

This corollary is an application of Theorem 18, where π_s is the identity map. In other literatures, e.g. [12], condition (ii) is called *normality* of the language \mathcal{S} with respect to the projection π_z .

Proposition 23 *Given a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , and a specification \mathcal{S} that is achievable. If π_c is replaced with any $\phi_c \succeq \pi_c$ such that the homogeneity property is still satisfied, then \mathcal{S} is still achievable.*

Proof. It is sufficient to show that condition (ii) in Theorem 18 remains satisfied, even if π_c is replaced with $\phi_c \succeq \pi_c$. Notice that in general $\mathcal{S} \subset \pi_s \pi_c^{-1} \pi_c \pi_s^{-1} \mathcal{S}$. So it remains to prove that

$$(\mathcal{S} \supset \pi_s \pi_c^{-1} \pi_c \pi_s^{-1} \mathcal{S}) \Rightarrow (\mathcal{S} \supset \pi_s \phi_c^{-1} \phi_c \pi_s^{-1} \mathcal{S}). \tag{22}$$

Since $\phi_c \succeq \pi_c$ implies $\phi_c^{-1} \phi_c \subset \pi_c^{-1} \pi_c$, (22) is satisfied and hence \mathcal{S} is still achievable. ■

There is an intuitive rationale behind this proposition. Replacing π_c with ϕ_c means allowing higher amount of information to be used in the interconnection between the plant and the controller, and hence making it easier to achieve the desired specification.

⁶The terms observable and controllable refer to the usage in [12].

3.3 Control problems with a tolerance gap

In both variants of control problems that we have seen above, the goal of the control problem is to achieve an given specification \mathcal{S} . In this subsection, we shall consider the variant where the requirement is relaxed, by requiring that the controlled behavior lies between two specification bounds, \mathcal{S}_r and \mathcal{S}_a . The idea is that \mathcal{S}_r is the minimal required behavior and that \mathcal{S}_a is the maximal allowed behavior [12]. To avoid ill-posed problems, it is always assumed that $\mathcal{S}_r \subset \mathcal{S}_a$.

When such a tolerance gap is present, the full interconnection control problem discussed in Subsection 3.1 becomes finding a \mathcal{C} such that

$$\mathcal{S}_r \subset (\mathcal{P} \parallel \mathcal{C}) \subset \mathcal{S}_a. \quad (23)$$

The following theorem states the conditions for solvability (i.e. existence of a solution for \mathcal{C}) of such problem. A proof is not included since it is trivial.

Theorem 24 *The control problem associated with (23) is solvable if and only if $\mathcal{S}_r \subset \mathcal{P}$.*

The problem becomes more interesting when some projections are involved. The partial interconnection control problem in the presence of a tolerance gap becomes finding a \mathcal{C} such that

$$\mathcal{S}_r \subset \pi_s \pi_c^{-1} (\pi_c \mathcal{P} \parallel \mathcal{C}) \subset \mathcal{S}_a. \quad (24)$$

A candidate controller that solves (24) is the *first canonical controller* as follows.

$$\mathcal{C}'_{\text{can}} = \{c \in \pi_c \mathcal{P} \mid \pi_s \pi_c^{-1} c \subset \mathcal{S}_a\}. \quad (25)$$

This canonical controller also possesses the property analog to the one described in Theorem 16.

Theorem 25 *The control problem (24) is solvable, if and only if $\mathcal{C}'_{\text{can}}$ solves it.*

Proof. We only need to prove the "only if" part. We use Proposition 15 to establish that $\mathcal{C}'_{\text{can}}$ achieves the maximal achievable behavior in \mathcal{S}_a . Suppose that $\mathcal{C}'_{\text{can}}$ does not solve the problem, then

$$\mathcal{S}_r \not\subset \pi_s \pi_c^{-1} (\pi_c \mathcal{P} \parallel \mathcal{C}'_{\text{can}}). \quad (26)$$

Since $\mathcal{C}'_{\text{can}}$ achieves the maximal achievable behavior in \mathcal{S}_a , (26) implies that the problem is not solvable. ■

If the plant \mathcal{P} possesses the homogeneity property with respect to π_c and π_s (see previous subsection), we again construct the *second canonical controller*, in a way analog to the construction in the previous subsection, as follows.

$$\mathcal{C}''_{\text{can}} = \pi_c \pi_s^{-1} (\pi_s \mathcal{P} \parallel \mathcal{S}_r). \quad (27)$$

Theorem 26 *For a plant \mathcal{P} satisfying the homogeneity property with respect to π_s and π_c , all the following statements are equivalent.*

- (i) *The control problem (24) is solvable.*
- (ii) *$\mathcal{S}_r \subset \pi_s \mathcal{P}$, and $\mathcal{S}_a \supset \pi_s \pi_c^{-1} \pi_c \pi_s^{-1} \mathcal{S}_r$.*
- (iii) *The second canonical controller $\mathcal{C}''_{\text{can}}$ solves the control problem.*

Proof. (i) \Rightarrow (ii) Since all achievable behaviors must be contained in $\pi_s\mathcal{P}$, necessarily $\mathcal{S}_r \subset \pi_s\mathcal{P}$. The second canonical controller is then (see (27)) $\mathcal{C}_{\text{can}}'' = \pi_c\pi_s^{-1}\mathcal{S}_r$. Suppose that $\mathcal{S}_a \not\subset \pi_s\pi_c^{-1}\pi_c\pi_s^{-1}\mathcal{S}_r$. We are going to show that this is a contradiction. The behavior achieved by the second canonical controller is $\mathcal{K}_{\text{can}}'' := \pi_s\pi_c^{-1}(\pi_c\mathcal{P} \parallel \pi_c\pi_s^{-1}\mathcal{S}_r)$. Since $\mathcal{S}_r \subset \pi_s\mathcal{P}$, we also have that $\pi_c\pi_s^{-1}\mathcal{S}_r \subset \pi_c\mathcal{P}$. Hence

$$\mathcal{K}_{\text{can}}'' = \pi_s\pi_c^{-1}\pi_c\pi_s^{-1}\mathcal{S}_r \not\subset \mathcal{S}_a. \quad (28)$$

From Proposition 19, and since $\mathcal{S}_r \subset \pi_s\mathcal{P}$, we know that $\mathcal{K}_{\text{can}}''$ is the minimal achievable behavior containing \mathcal{S}_r . Therefore (28) implies that the control problem is not solvable.

(ii) \Rightarrow (iii) From the previous paragraph, it follows that if (ii) is satisfied then $\mathcal{S}_r \subset \mathcal{K}_{\text{can}}'' \subset \mathcal{S}_a$.

(iii) \Rightarrow (i) Trivial. ■

An example of control problems of this type can be obtained, for example by replacing the control problem in Corollary 22 with the version with a tolerance gap. In this case, \mathcal{S}_r is present to make sure that the supervised language possess some "liveliness" property, and \mathcal{S}_a is to prevent some undesirable executions to take place (for example, for safety reasons).

3.4 Constrained control problems

In all the variants of control problems discussed so far, we only have to find a controller behavior that satisfies the control objective. No constraints are present to restrict our choice. However, in most cases, the problems are presented together with some constraints inherent to them.

Some examples of the constraints are:

- **Structural constraint**, where the controller behavior is required to be of a certain structure. For example, not any collection of continuous trajectories are allowed, but it must also be linear time invariant, or not any collection of all strings are allowed, but it must also be prefix closed, etc.
- **Compatibility constraint**, where the controller behavior is required to meet some compatibility criteria with respect to the plant.

Let us discuss the structural constraint from the point of view of the discussion so far. Two controllers have been introduced, namely the *first and second canonical* controller. The construction of the first canonical controller is purely based on set theoretic approach. The construction of the second canonical controller, although based on set theoretic approach as well, uses composition of behaviors. Thus, given that the plant and the specification belong to the appropriate structure, as required by the structural constraint, and that the structure is closed under composition, the second canonical controller automatically satisfies the constraint as well. For example, if both the plant and the specification are linear time invariant behaviors, we know that the second canonical controller is also a linear time invariant behavior. In addition, linear time invariant behaviors with projections as described in Example 4 satisfy the *homogeneity* property, thus the idea of second canonical controller can be used.

Compatibility constraint is present, for example because of some *input-output* structure **inherent** to the plant. We stress on the inherence of the input-output structure to the plant,

because in many systems, although the interconnection of the plant and the controller does possess some input-output structure, this structure is not inherent to the plant. The structure belongs to the interconnection, rather than the plant. We refer the reader to [2] for an interesting discussion on this issue. Compatibility constraints corresponding to input-output structure appear, for example in [8], where compatibility of hybrid input-output automata is defined based on the matching of inputs and outputs of the automata.

Compatibility constraint also arises when we consider the causality of the effect of the interconnection. The main idea is as follows. Take two behaviors \mathfrak{B}_1 and \mathfrak{B}_2 of type (\mathbb{T}, \mathbb{W}) . Recall that we require that the time axis is totally ordered. Since time instants are ordered, it makes sense to talk about causality. From Definition 8 of the composition of behaviors, it is clear that the composed behavior $\mathfrak{B}_1 \parallel \mathfrak{B}_2$ is smaller than both \mathfrak{B}_1 and \mathfrak{B}_2 separately. This means that in general there will be trajectories of \mathfrak{B}_1 and \mathfrak{B}_2 that are not "accepted" in the composed behavior.

In some cases, for example in discrete event systems, it is desired to assume that the composition is formed at the infimum of \mathbb{T} . The rejected trajectories then do not give rise to any conflict, since it is assumed that they are prevented from happening from the beginning of time. In other cases, it is necessary to assume that the interconnection is formed within \mathbb{T} , say at time $\tau \in \mathbb{T}$. The causality of the interconnection then requires that any forbidden trajectory can switch to an accepted one before or at time τ . If this is not met, then the interconnection is not compatible, since it leads to undesired consequences such as deadlock (for discrete event systems) or that the interconnection is ill-defined (e.g. nonregular interconnection of linear behaviors). This type of compatibility is discussed in more detail in [11].

4 Concluding remarks

We have presented a general mathematical framework for interconnected systems. The framework is based on the behavioral approach to systems theory.

The main part of the framework is that interconnections or compositions of systems can be modelled with interconnection of behaviors and projection operations. We also discussed control problems from the point of view of behavioral interconnections. Several variants of the problem have been presented, together with the solutions in terms of the construction of first and second canonical controller.

At the end of the discussion, we discussed about constrained control problems, a class in which many real control problems fall. Of the constraints discussed, compatibility constraints received more attention. Although not explicitly explained, this type of constraints can also be expressed in terms of projections and behavioral inclusions. Therefore, we think that the next step in working with the framework is to extend it to cover these items. In particular, we aim at solving control problems expressed as conjunctions of many subproblems of the type expressed in (15) and (24).

References

- [1] Willems, J.C.: Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. Automatic Control* **36(3)** (1991) 259–294

- [2] Willems, J.C.: On interconnections, control and feedback. *IEEE Trans. Automatic Control* **42(3)** (1997) 326–339
- [3] Polderman, J.W., Willems, J.C.: *Introduction to Mathematical Systems Theory : A Behavioral Approach*. Springer, New York (1998)
- [4] Belur, M.N.: *Control in a Behavioral Context*. PhD thesis, Rijksuniversiteit Groningen (2003)
- [5] Smedinga, R.: *Control of Discrete Events*. PhD thesis, Rijksuniversiteit Groningen (1989)
- [6] Moor, T., Raisch, J.: Supervisory control of hybrid systems within a behavioral framework. *Systems and Control Letters* **38(3)** (1999) 157–166
- [7] Julius, A.A., Strubbe, S.N., van der Schaft, A.J.: Control of hybrid behavioral automata by interconnection. In: *Proc. IFAC Conf. Analysis and Design of Hybrid Systems*, St. Malo, IFAC (2003) 135–140
- [8] Lynch, N.A., Segala, R., Vandraager, F.W.: Hybrid I/O automata. *Information and Computation* **185(1)** (2003) 105–157
- [9] van der Schaft, A.J., Julius, A.A.: Achievable behavior by composition. In: *Proceedings 41st IEEE Conf. Decision and Control, Las Vegas, IEEE* (2002) 7–12
- [10] van der Schaft, A.J.: Achievable behavior of general systems. *Systems and Control Letters* **49** (2003) 141–149
- [11] Julius, A.A., van der Schaft, A.J.: Compatibility of behavior interconnections. In: *Proc. European Control Conf., Cambridge, UK, IEE* (2003)
- [12] Cassandras, C.G., Lafortune, S.: *Introduction to Discrete Event Systems*. Kluwer (1999)
- [13] Trentelman, H.L., Willems, J.C.: H-infinity control in a behavioral context: The full information case. *IEEE Trans. Automatic Control* **44(3)** (1999) 521–536
- [14] Willems, J.C., Belur, M.N., Julius, A.A., Trentelman, H.L.: The canonical controller and its regularity. In: *Proc. IEEE Conference on Decision and Control, Hawaii, IEEE* (2003) 1639–1644