Behavioural assume-guarantee contracts for linear dynamical systems

Shali, Brayan; van der Schaft, Arjan; Besselink, Bart

[Link to publication in University of Groningen/UMCG research database](#)

Behavioural assume-guarantee contracts for linear dynamical systems

B. M. Shali, A. J. van der Schaft, B. Besselink

Abstract—**Motivated by the growing requirements on the operation of complex engineering systems, we present contracts as specifications for continuous-time linear dynamical systems with inputs and outputs. A contract is defined as a pair of assumptions and guarantees, both characterized in a behavioural framework. The assumptions encapsulate the available information about the dynamic behaviour of the environment in which the system is supposed to operate, while the guarantees express the desired dynamic behaviour of the system when interconnected with relevant environments. In addition to defining contracts, we characterize contract implementation, and we find necessary conditions for the existence of an implementation. We also characterize contract refinement, which is used to characterize contract conjunction in two special cases. These concepts are then illustrated by an example of a vehicle following system.**

I. INTRODUCTION

Contemporary engineering systems are becoming increasingly more complex. They often comprise a large number of interconnected components, which can be quite complex themselves, thus making the design and analysis of the overall system prohibitively difficult. One way to circumvent this problem is by employing a method for design and analysis that is inherently *modular*, i.e., that allows components to be considered independently. Contract-based design, which finds its origins in the field of software engineering [1], is precisely such a method.

Motivated by this, we present assume-guarantee contracts for continuous-time linear dynamical systems with inputs and outputs. There are many different types of contract theories depending on the context in which they are developed [2]. Nevertheless, they all share a common objective, namely, to support the independent design of components within interconnected systems. This is typically done by defining appropriate concepts, such as refinement and composition, that satisfy certain compositionality properties. The concepts and philosophy behind contract theories are captured in a mathematical meta-theory of contracts presented in [2].

The contracts in this paper provide an alternative to common methods for expressing system specifications in control, such as dissipativity [3] and set-invariance [4]. While the latter represent specifications that are typically static, e.g., static supply rates and invariant sets, the contracts in this paper are used to specify dynamic behaviour. Furthermore, the contracts in this paper express specifications directly in the continuous domain, unlike formal methods in control [5],

The authors are with the Jan C. Willems Center for Systems and Control, and the Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, University of Groningen, Groningen, The Netherlands; Email: *b.m.shali@rug.nl*; *a.j.van.der.schaft@rug.nl*; *b.besselink@rug.nl*.

which usually require the abstraction to discrete transition systems in order to express specifications using LTL.

We make the following contributions in this paper. First, we define assume-guarantee contracts for linear systems, where the assumptions and guarantees are linear systems that represent an expected input behaviour and a desired external behaviour, respectively. These contracts allow us to express rich specifications directly in the continuous domain. Second, we define and characterize contract implementation, which leads to necessary conditions for contract consistency, i.e., the existence of an implementation. Third, we define and characterize contract refinement, which we then use to define contract conjunction and characterize it in two special cases.

The contracts presented in this paper are a generalization of the contracts in [6], and are closely related to the contracts in [7]. In contrast to [6], where the guarantees specify only a set of admissible output trajectories, the guarantees in this paper can also specify a particular relationship between input and output trajectories. As such, the contracts in this paper truly specify input-output behaviour and enable the expression of significantly richer specifications than [6]. At the same time, the developed theory is based on the equations representing linear systems, not on their solutions, which leads to conditions that are much easier to verify.

Different types of contracts have already been used as specifications for dynamical systems. A small-gain theorem for parametric assume-guarantee contracts is presented in [8] and used for controller synthesis in [9]. These contracts are also adopted in [10] for safety-critical control synthesis in network systems. On the other hand, assume-guarantee contracts that capture invariance properties are presented in [11] and used for symbolic controller synthesis in [12], [13]. Applications of these contracts can be found in [14], [15]. Contracts for safety are presented in [16], while the contracts in [17] are used for finite-time reach and avoid or infinite-time invariance. Finally, the authors of [18] focus on using contracts to represent linear constraints, and develop efficient computational tools based on linear programming. As opposed to the contracts in this paper, which are used to express specifications on the dynamics of continuous-time systems, the contracts in [8], [9], [10], [18] are defined only for discrete-time systems, and the contracts in [11], [12], [13], [14], [15], [16] cannot express specifications on dynamics.

The remainder of this paper is structured as follows. In Section II, we introduce the class of systems that will be considered in this paper. Contracts are introduced in Section III, where we define and characterize contract implementation, and discuss contract consistency. Contract refinement and

conjunction are treated in Section IV. We demonstrate the theoretical framework with an illustrative example in Section V, followed by a conclusion in Section VI.

The notation in this paper is mostly standard. We denote the space of smooth functions from $\mathbb{R}$ to $\mathbb{R}^n$ by $\mathcal{C}_n^\infty$. A *polynomial matrix* is a matrix whose entries are polynomials, and a *rational matrix* is a matrix whose entries are rational functions. Throughout this paper, all polynomials are univariate and have real coefficients. A rational function is *proper* if the degree of its denominator is greater than or equal to the degree of its numerator, and a rational matrix is proper if all of its entries are proper rational functions. A square polynomial matrix $P(s)$ is called *invertible* if there exists a rational matrix $Q(s)$ such that $P(s)Q(s) = I$. If there exists a *polynomial* matrix $Q(s)$ such that $P(s)Q(s) = I$, then $P(s)$ is called *unimodular*. In both cases, the matrix $Q(s)$ is referred to as the *inverse* of $P(s)$ and is denoted by $P(s)^{-1}$.

II. MODELS OF PHYSICAL SYSTEMS

In this paper, we will consider systems of the form

$$\Sigma : \begin{cases} \dot{x} = Ax + Bu, \\ y = Cx + Du, \end{cases} \qquad (1)$$

where $x \in \mathcal{C}_n^\infty$ is the state trajectory, $u \in \mathcal{C}_m^\infty$ is the input trajectory, and $y \in \mathcal{C}_p^\infty$ is the output trajectory. We regard u and y as external variables that can interact with the environment, while x is internal, as illustrated in Figure 1.
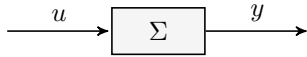

Fig. 1. The system Σ as a signal processor.

Since we are interested in the interaction of a system with its environment, we consider the *external behaviour*

$$\mathfrak{B}(\Sigma) = \left\{ (u,y) \in \mathcal{C}_{m+p}^\infty \mid \exists x \in \mathcal{C}_n^\infty \text{ s.t. (1) holds} \right\}. \quad (2)$$

In the behavioural approach to systems theory [19], [20], the system Σ is seen as a representation of its external behaviour $\mathfrak{B}(\Sigma)$. In view of [21, Theorem 6.2], the same external behaviour can always be represented by a system of the form

$$\Sigma : P\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)y = Q\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)u, \qquad (3)$$

where $u \in \mathcal{C}_m^\infty$, $y \in \mathcal{C}_p^\infty$, and $P(s)$ and $Q(s)$ are polynomial matrices such that $P(s)$ is invertible and $P(s)^{-1}Q(s)$ is proper. If these conditions on $P(s)$ and $Q(s)$ are satisfied, then we say that Σ of the form (3) is in *input-output form* [19, Section 3.3]. As (3) involves only the external variables u and y, we will represent external behaviours by systems Σ of the form (3) in input-output form.

The concept of behaviour allows one to compare different systems. In particular, if $\mathfrak{B}(\Sigma_1) \subset \mathfrak{B}(\Sigma_2)$, then for a given input trajectory $u \in \mathcal{C}_m^\infty$, the set of output trajectories produced by Σ_1 is contained in the set of output trajectories produced by Σ_2, hence Σ_2 can be interpreted as having richer dynamics than Σ_1. Behavioural inclusion plays a major role in the definition of a contract and its related concepts. The

following theorem, whose proof can be found in [6], see also [22], provides an algebraic characterization of behavioural inclusion that will be used in the following sections.

Theorem 1 ([6, Theorem 2]): Consider the behaviours

$$\mathfrak{B}_j = \left\{ w \in \mathcal{C}_k^\infty \mid R_j\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)w = 0 \right\}, \ j \in \{1,2\},$$

where $R_1(s)$ and $R_2(s)$ are polynomial matrices. Then $\mathfrak{B}_1 \subset \mathfrak{B}_2$ if and only if there exists a polynomial matrix $M(s)$ such that $R_2(s) = M(s)R_1(s)$.

III. CONTRACTS

In this section, we will define contracts and show how they can be used as specifications. Consider a system Σ of the form (3) in input-output form. As an open system, Σ typically operates in interconnection with its environment, which is responsible for providing input trajectories. In view of this, we define an *environment* E as a system of the form

$$\text{E} : 0 = E\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)u, \qquad (4)$$

where $u \in \mathcal{C}_m^\infty$ and $E(s)$ is a polynomial matrix. The environment E defines the *input behaviour*

$$\mathfrak{B}_\mathrm{i}(\text{E}) = \left\{ u \in \mathcal{C}_m^\infty \mid (4) \text{ holds} \right\},$$

and the interconnection of Σ with E is obtained by setting the input generated by E as input of Σ. This yields the system

$$\text{E} \wedge \Sigma : \begin{bmatrix} P\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ 0 \end{bmatrix} y = \begin{bmatrix} Q\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ E\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} u, \qquad (5)$$

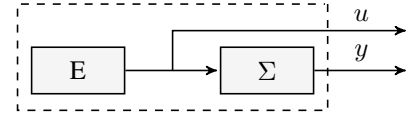which is represented graphically in Figure 2.

Fig. 2. The interconnection $\text{E} \wedge \Sigma$.

As a design goal, we are interested in guaranteeing properties of the external behaviour $\mathfrak{B}(\text{E} \wedge \Sigma)$. Since this is partially determined by the environment E, any available information about E can ease the design burden on the system Σ and should thus be taken into account. To formalize this, we will introduce another two systems. First, the assumptions A are a system of the form

$$\text{A} : 0 = A\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)u, \qquad (6)$$

where $u \in \mathcal{C}_m^\infty$ and $A(s)$ is a polynomial matrix. Just like an environment, the assumptions A represent the input behaviour $\mathfrak{B}_\mathrm{i}(\text{A})$. Second, the guarantees Γ are a system of the form

$$\Gamma : G\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)y = H\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right)u, \qquad (7)$$

where $u \in \mathcal{C}_m^\infty$, $y \in \mathcal{C}_p^\infty$, and $G(s)$ and $H(s)$ are polynomial matrices. The guarantees Γ represent the external behaviour $\mathfrak{B}(\Gamma)$, just like the interconnection of a system with its environment. Then contracts are defined as follows.

Definition 1: A *contract* $\mathcal{C}$ is a pair (A, Γ) of assumptions A and guarantees Γ.

The interpretation of a contract is given in the following definition.

Definition 2: An environment E is *compatible* with the contract $\mathcal{C} = (A, \Gamma)$ if $\mathfrak{B}_i(E) \subset \mathfrak{B}_i(A)$. A system Σ of the form (3) in input-output form *implements* $\mathcal{C}$ if $\mathfrak{B}(E \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ for all environments E compatible with $\mathcal{C}$. In such a case, we say that Σ is an *implementation* of $\mathcal{C}$.

In other words, the assumptions capture the available information about the environment, thus leading to a class of compatible environments, while the guarantees represent the desired behaviour of the system when interconnected with any compatible environment, thus leading to a class of implementations. These two aspects of a contract constitute a formal specification for the external behaviour of a system.

Remark 1: The assume-guarantee contracts presented in this paper are very closely related to the assume-guarantee contracts presented in [6]. The main difference is in the guarantees, which represent an external behaviour in this paper, rather than an output behaviour only, as in [6]. This means that the contracts presented in this paper specify a *relationship* between inputs and outputs, not only a set of admissible outputs. In any case, the contracts in [6] are a special case of the contracts here, obtained by restricting guarantees to be of the form (7) with $H(s) = 0$.

Definition 1 suggests that checking contract implementation requires the construction of all compatible environments. The following theorem shows that this is not necessary.

Theorem 2: A system Σ as in (3) in input-output form implements $\mathcal{C} = (A, \Gamma)$ if and only if $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$.

Proof: Suppose that $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ and let E be an environment compatible with $\mathcal{C}$. If $(u, y) \in \mathfrak{B}(E \wedge \Sigma)$ then $u \in \mathfrak{B}_i(E) \subset \mathfrak{B}_i(A)$ and $(u, y) \in \mathfrak{B}(\Sigma)$, hence $(u, y) \in \mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$. This shows that $\mathfrak{B}(E \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ for all compatible environments E, thus Σ implements $\mathcal{C}$. Conversely, suppose that Σ is an implementation of $\mathcal{C}$. Then $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ because A is compatible with $\mathcal{C}$. ∎

Remark 2: Suppose that Σ, A and Γ are given by (3), (6) and (7), respectively. Using Theorem 1 with $w = [y^\top \quad u^\top]^\top$ shows that $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ if and only if there exist polynomial matrices $M_1(s)$ and $M_2(s)$ such that

$$\begin{bmatrix} G(s) & -H(s) \end{bmatrix} = \begin{bmatrix} M_1(s) & M_2(s) \end{bmatrix} \begin{bmatrix} P(s) & -Q(s) \\ 0 & -A(s) \end{bmatrix}.$$

The existence of such polynomial matrices $M_1(s)$ and $M_2(s)$ can be verified using [6, Lemma 1].

In practice, we are only interested in contracts that can be implemented, hence the following definition.

Definition 3: A contract $\mathcal{C}$ is *consistent* if there exists at least one implementation of $\mathcal{C}$.

Not all contracts are consistent. One reason is that implementations cannot restrict the input u, hence any restrictions on u imposed by the guarantees must already be present in the assumptions. To formalize this, we introduce the following definition.

Definition 4: Given a system that involves both u and y, its *input behaviour* is defined as the projection of its external behaviour onto u. For guarantees Γ, this means that

$$\mathfrak{B}_i(\Gamma) = \left\{ u \in \mathcal{C}_m^\infty \;\middle|\; \exists y \in \mathcal{C}_p^\infty \text{ s.t. } (u, y) \in \mathfrak{B}(\Gamma) \right\}. \quad (8)$$

Using Definition 4 we obtain the following necessary condition for consistency.

Lemma 3: The contract $\mathcal{C} = (A, \Gamma)$ is consistent only if $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$. In this case, there exist guarantees

$$\Gamma' : G'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) y = H'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) u, \quad (9)$$

such that $G'(s)$ has full row rank and $\mathfrak{B}(A \wedge \Gamma) = \mathfrak{B}(A \wedge \Gamma')$.

Proof: Suppose that $\mathcal{C} = (A, \Gamma)$ is consistent. In view of Theorem 2, this implies that there exists Σ as in (3) in input-output form such that $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$. Since Σ is in input-output form, it follows that u is free in $\mathfrak{B}(\Sigma)$, i.e., for all $u \in \mathcal{C}_m^\infty$, there exists $y \in \mathcal{C}_p^\infty$ such that $(u, y) \in \mathfrak{B}(\Sigma)$. Consequently, we obtain $\mathfrak{B}_i(A \wedge \Sigma) = \mathfrak{B}_i(A)$, hence $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ only if $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$.

We proceed with finding guarantees Γ' of the form (9) such that $G'(s)$ has full row rank and $\mathfrak{B}(A \wedge \Gamma') = \mathfrak{B}(A \wedge \Gamma)$. Let Γ be given by (7). If $G(s)$ has full row rank, then we can take $\Gamma' = \Gamma$. Now, suppose that $G(s)$ does not have full row rank. In light of [19, Theorem 6.2.6], there exists a unimodular matrix $U(s)$ such that

$$U(s)G(s) = \begin{bmatrix} G'(s) \\ 0 \end{bmatrix}, \quad U(s)H(s) = \begin{bmatrix} H'(s) \\ H''(s) \end{bmatrix}, \quad (10)$$

and $G'(s)$ has full row rank. From the same theorem, we also know that the input behaviour $\mathfrak{B}_i(\Gamma)$ is given by

$$\mathfrak{B}_i(\Gamma) = \left\{ u \in \mathcal{C}_m^\infty \;\middle|\; H''\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) u = 0 \right\}.$$

Since $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$, Theorem 1 implies that there exists a polynomial matrix $M(s)$ such that $H''(s) = M(s)A(s)$. We claim that the guarantees Γ' given by (9), where $G'(s)$ and $H'(s)$ are given by (10), are such that $\mathfrak{B}(A \wedge \Gamma) = \mathfrak{B}(A \wedge \Gamma')$. To see this, note that $(u, y) \in \mathfrak{B}(A \wedge \Gamma)$ if and only if

$$\begin{bmatrix} G'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ 0 \\ 0 \end{bmatrix} y = \begin{bmatrix} H'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ H''\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ A\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} u \quad (11)$$

because $U(s)$ is unimodular and (10) holds, see [19, Theorem 2.5.4]. But we have that $H''(s) = M(s)A(s)$, hence (11) holds if and only if

$$\begin{bmatrix} G'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ 0 \end{bmatrix} y = \begin{bmatrix} H'\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \\ A\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\right) \end{bmatrix} u. \quad (12)$$

Since (12) holds if and only if $(u, y) \in \mathfrak{B}(A \wedge \Gamma')$, we conclude that $\mathfrak{B}(A \wedge \Gamma) = \mathfrak{B}(A \wedge \Gamma')$, as desired. ∎

The condition that $G'(s)$ has full row rank ensures that Γ' does not restrict u, see the proof of [19, Theorem 6.2.6]. In view of Theorem 2 and the fact that $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(\Gamma)$ if and only if $\mathfrak{B}(A \wedge \Sigma) \subset \mathfrak{B}(A \wedge \Gamma)$, it follows that the contracts $\mathcal{C} = (A, \Gamma)$ and $\mathcal{C}' = (A, \Gamma')$ define the same class of implementations. In other words, Lemma 3 tells us that the guarantees of a consistent contract can be replaced by guarantees that do not restrict u.

Unfortunately, the inclusion $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$ is not sufficient for the contract $\mathcal{C} = (A, \Gamma)$ to be consistent, as illustrated by the following example.

Example 1: Consider assumptions A given by (6) with $A(s) = 0$, and guarantees Γ be given by (7) with $G(s) = 1$ and $H(s) = s$. It is easily seen that $\mathfrak{B}_i(A) = \mathcal{C}_1^\infty = \mathfrak{B}_i(\Gamma)$, and in particular, that $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$. Nevertheless, we will show that $\mathcal{C} = (A, \Gamma)$ is not consistent. Let the system Σ be of the form (3) in input-output form. In view of Remark 2, we have that Σ implements $\mathcal{C}$ if and only if there exists a polynomial matrices $M_1(s)$ and $M_2(s)$ such that

$$\begin{bmatrix} 1 & -s \end{bmatrix} = \begin{bmatrix} M_1(s) & M_2(s) \end{bmatrix} \begin{bmatrix} P(s) & -Q(s) \\ 0 & 0 \end{bmatrix}.$$

But the latter holds only if $M_1(s) \neq 0$ and $P(s)^{-1}Q(s) = s$, which contradicts the assumption that Σ is in input-output form. It follows that $\mathcal{C}$ does not have an implementation and thus it is not consistent.

At first sight, it might seem that the inconsistency of $\mathcal{C}$ is caused by the fact that the guarantees Γ are not in input-output form, or more precisely, that $G(s)^{-1}H(s)$ is not proper. This is only partially true. While it is true that $\mathcal{C}$ would be consistent if the guarantees Γ were in input-output form ($\Sigma = \Gamma$ would be an implementation), this is generally not necessary. Indeed, if the assumptions A were given by (6) with $A(s) = s$ and the guarantees Γ were the same, then $\mathcal{C}$ would be consistent because the system Σ given by (3) with $P(s) = 1$ and $Q(s) = 0$ would be an implementation. The latter follows from Remark 2 and the fact that

$$\begin{bmatrix} 1 & -s \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -s \end{bmatrix}.$$

In other words, even when $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$, consistency still depends on both the assumptions A and the guarantees Γ.

IV. REFINEMENT AND CONJUNCTION

A central concept in any contract theory is the concept of contract refinement. Refinement allows one to compare contracts, which has an essential role in enabling the independent design of components. Following the meta-theory in [2], we define contract refinement as follows.

Definition 5: The contract $\mathcal{C}_1$ *refines* the contract $\mathcal{C}_2$, denoted by $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$, if the following conditions hold:
1) all compatible environments of $\mathcal{C}_2$ are compatible environments of $\mathcal{C}_1$;
2) all implementations of $\mathcal{C}_1$ are implementations of $\mathcal{C}_2$.

Said differently, $\mathcal{C}_1$ refines $\mathcal{C}_2$ if it specifies stricter guarantees that have to be satisfied under weaker assumptions, i.e., for a larger class of compatible environments. Then $\mathcal{C}_1$ can be seen as expressing a tighter specification than $\mathcal{C}_2$. Just like implementation, we can characterize refinement on the basis of assumptions and guarantees alone. If $\mathcal{C}_1 = (A_1, \Gamma_1)$ and $\mathcal{C}_2 = (A_2, \Gamma_2)$, then it is not difficult to see that the first condition in Definition 5 holds if and only if $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$. For the second condition, we need to consider two cases depending on whether $\mathcal{C}_1$ is consistent. If $\mathcal{C}_1$ is not consistent, then the second condition is vacuously satisfied,

hence $\mathcal{C}_1$ refines $\mathcal{C}_2$ if and only if $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$. If $\mathcal{C}_1$ is consistent, then we have the following theorem, whose proof can be found in the appendix.

Theorem 4: If the contract $\mathcal{C}_1 = (A_1, \Gamma_1)$ is consistent, then it refines the contract $\mathcal{C}_2 = (A_2, \Gamma_2)$ if and only if $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$ and $\mathfrak{B}(A_2 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_2)$. In this case, the contract $\mathcal{C}_2$ is guaranteed to be consistent.

While refinement allows us to compare contracts, the concept of conjunction allows us to combine contracts. In particular, if a contract $\mathcal{C}$ refines another two contracts $\mathcal{C}_1$ and $\mathcal{C}_2$, then $\mathcal{C}$ can be interpreted as enforcing both $\mathcal{C}_1$'s and $\mathcal{C}_2$'s specification. This motivates the following definition.

Definition 6: The *conjunction* of contracts $\mathcal{C}_1$ and $\mathcal{C}_2$, denoted by $\mathcal{C}_1 \wedge \mathcal{C}_2$, is the largest (with respect to contract refinement) contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$.

Remark 3: In Definition 6, by $\mathcal{C}_1 \wedge \mathcal{C}_2$ being the largest contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, we mean that $\mathcal{C}_1 \wedge \mathcal{C}_2$ refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, and is refined by any contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$. Note that such a contract does not necessarily exist.

The definition of the conjunction $\mathcal{C}_1 \wedge \mathcal{C}_2$ has two aspects. On the one hand, $\mathcal{C}_1 \wedge \mathcal{C}_2$ is a contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, hence it expresses a fusion of the specifications that $\mathcal{C}_1$ and $\mathcal{C}_2$ express. On the other hand, $\mathcal{C}_1 \wedge \mathcal{C}_2$ is the *largest* such contract, hence it expresses the *least restrictive* fusion of the specifications that $\mathcal{C}_1$ and $\mathcal{C}_2$ express.

Considering the first aspect of the conjunction $\mathcal{C}_1 \wedge \mathcal{C}_2$, we can always find a contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$.

Lemma 5: If $\mathcal{C}_1 = (A_1, \Gamma_1)$ and $\mathcal{C}_2 = (A_2, \Gamma_2)$, then

$$\mathcal{C} = (A_1 \vee A_2, \Gamma_1 \wedge \Gamma_2)$$

refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, where

$$A_1 \vee A_2 : \begin{bmatrix} I & I \\ A_1\left(\frac{d}{dt}\right) & 0 \\ 0 & A_2\left(\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} u,$$

is the *join* of A_1 and A_2, and

$$\Gamma_1 \wedge \Gamma_2 : \begin{bmatrix} G_1\left(\frac{d}{dt}\right) \\ G_2\left(\frac{d}{dt}\right) \end{bmatrix} y = \begin{bmatrix} H_1\left(\frac{d}{dt}\right) \\ H_2\left(\frac{d}{dt}\right) \end{bmatrix} u$$

is the *meet* of Γ_1 and Γ_2.

Proof: The join $A_1 \vee A_2$ is such that $u \in \mathfrak{B}_i(A_1 \vee A_2)$ if and only if $u = l_1 + l_2$ for $l_1 \in \mathfrak{B}_i(A_1)$ and $l_2 \in \mathfrak{B}_i(A_2)$, while the meet $\Gamma_1 \wedge \Gamma_2$ is such that $(u, y) \in \mathfrak{B}(\Gamma_1 \wedge \Gamma_2)$ if and only if $(u, y) \in \mathfrak{B}(\Gamma_1)$ and $(u, y) \in \mathfrak{B}(\Gamma_2)$. Therefore, we have that $\mathfrak{B}_i(A_1 \vee A_2) = \mathfrak{B}_i(A_1) + \mathfrak{B}_i(A_2)$ and $\mathfrak{B}(\Gamma_1 \wedge \Gamma_2) = \mathfrak{B}(\Gamma_1) \cap \mathfrak{B}(\Gamma_2)$. The former immediately implies that every environment compatible with $\mathcal{C}_1$ or $\mathcal{C}_2$ is also compatible with $\mathcal{C}$, hence the first condition for refinement is satisfied. Note that if $\mathcal{C}$ is not consistent, then it has no implementations and the second condition for refinement is vacuously satisfied. With this in mind, suppose that $\mathcal{C}$ is consistent. Let Σ be an implementation and note that $\mathfrak{B}((A_1 \vee A_2) \wedge \Sigma) \subset \mathfrak{B}(\Gamma_1 \wedge \Gamma_2)$ due to Theorem 2. Since $\mathfrak{B}(A_i \wedge \Sigma) \subset \mathfrak{B}((A_1 \vee A_2) \wedge \Sigma)$ and $\mathfrak{B}(\Gamma_1 \wedge \Gamma_2) \subset \mathfrak{B}(\Gamma_i)$ for $i \in \{1, 2\}$, it follows that $\mathfrak{B}_i(A_i \wedge \Sigma) \subset \mathfrak{B}_i(\Gamma_i)$, hence Σ

implements $\mathcal{C}_i$ due to Theorem 2. This shows that the second condition for refinement is satisfied and we conclude that $\mathcal{C}$ refines both $\mathcal{C}_1$ and $\mathcal{C}_2$. ∎

Remark 4: Although we have defined $A_1 \vee A_2$ with the help of the *latent variables* l_1 and l_2, due to [19, Theorem 6.2.6], we can eliminate these to represent $\mathfrak{B}_i(A_1 \vee A_2)$ by a system of the form (6), like any other assumptions.

While we can always find a contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, it is not clear whether a *largest* such contract exists. In [6] it is shown that the conjunction of $\mathcal{C}_1 = (A_1, \Gamma_1)$ and $\mathcal{C}_2 = (A_2, \Gamma_2)$ exists when Γ_1 and Γ_2 are of the form (7) with $H(s) = 0$. In such a case, we have that

$$\mathcal{C}_1 \wedge \mathcal{C}_2 = (A_1 \vee A_2, \Gamma_1 \wedge \Gamma_2). \tag{13}$$

The following theorem, whose proof can be found in the appendix, shows that the conjunction $\mathcal{C}_1 \wedge \mathcal{C}_2$ exists and has the same form in another two special cases.

Theorem 6: The conjunction of contracts $\mathcal{C}_1 = (A_1, \Gamma_1)$ and $\mathcal{C}_2 = (A_2, \Gamma_2)$ is given by (13) if at least one of the following conditions holds:

1) $\mathfrak{B}_i(A_1) = \mathfrak{B}_i(A_2)$;
2) $\mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_1) = \mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_2)$.

Note that the first condition in Theorem 6 is that $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same assumptions, which implies that

$$\mathcal{C}_1 \wedge \mathcal{C}_2 = (A_1, \Gamma_1 \wedge \Gamma_2) = (A_2, \Gamma_1 \wedge \Gamma_2).$$

On the other hand, the second condition is that $\mathcal{C}_1$ and $\mathcal{C}_2$ have the same guarantees when restricted to inputs in the assumptions of $\mathcal{C}_1 \wedge \mathcal{C}_2$, hence

$$\mathcal{C}_1 \wedge \mathcal{C}_2 = (A_1 \vee A_2, \Gamma_1) = (A_1 \vee A_2, \Gamma_2).$$

V. ILLUSTRATIVE EXAMPLE

In this section, we will show how contracts can be used as specifications in the context of a vehicle following system. To this end, consider two vehicles, the leader E and the follower Σ, which are shown graphically in Figure 3. We
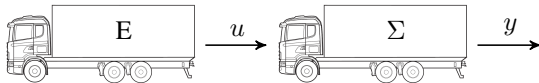
Fig. 3. A vehicle following system.

assume that the positions p_E, p_Σ, and velocities v_E, v_Σ, are external variables that are available for measurement. Our goal is to design Σ in such a way as to guarantee tracking of the constant headway spacing policy

$$p_E - p_\Sigma = h v_\Sigma, \tag{14}$$

where $h > 0$ is a constant parameter. By tracking, we mean that $p_E - p_\Sigma - h v_\Sigma$ converges to 0 asymptotically. Note that this would be the case if

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_E - p_\Sigma - h v_\Sigma) = -k(p_E - p_\Sigma - h v_\Sigma) \tag{15}$$
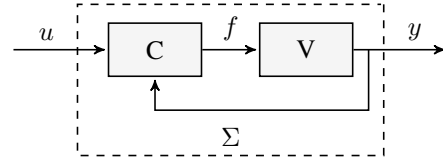
for some constant $k > 0$.

Fig. 4. The controlled vehicle Σ.

We can express this specification by a contract. We take the input of the follower vehicle Σ to be $u = [p_E \quad v_E]^\top$, and the output to be $y = [p_\Sigma \quad v_\Sigma]^\top$. Regardless of the particular dynamics of the leading vehicle E, it is safe to assume that v_E is the derivative of p_E, hence the corresponding assumptions A are given by (6) with

$$A(s) = \begin{bmatrix} s & -1 \end{bmatrix}.$$

On the other hand, in view of (15), the corresponding guarantees Γ are given by (7) with

$$G(s) = \begin{bmatrix} s+k & hs+hk \end{bmatrix}, \quad H(s) = \begin{bmatrix} s+k & 0 \end{bmatrix},$$

Then Σ tracks the spacing policy (in the sense of (15)) if Σ implements the contract $\mathcal{C} = (A, \Gamma)$.

We now turn to verifying that a given controlled vehicle Σ implements $\mathcal{C}$. To this end, we will model the follower vehicle as a simple mechanical system with unit mass and dynamics given by

$$V : \begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} & -1 \\ 0 & \frac{\mathrm{d}}{\mathrm{d}t} \end{bmatrix} y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} f,$$

where f is the force that will be provided by the controller. We claim that the controller

$$C : f = \begin{bmatrix} h^{-1}k & h^{-1} \end{bmatrix} u - \begin{bmatrix} h^{-1}k & h^{-1}+k \end{bmatrix} y$$

ensures that the controlled vehicle implements the contract $\mathcal{C} = (A, \Gamma)$. To show this, note that the dynamics of the controlled vehicle $\Sigma = V \wedge C$, shown in Figure 4, are given by (3) with

$$P(s) = \begin{bmatrix} s & -1 \\ h^{-1}k & s+h^{-1}+k \end{bmatrix}, \quad Q(s) = \begin{bmatrix} 0 & 0 \\ h^{-1}k & h^{-1} \end{bmatrix}.$$

We can easily verify that

$$\begin{bmatrix} G(s) & -H(s) \end{bmatrix} = \begin{bmatrix} M_1(s) & M_2(s) \end{bmatrix} \begin{bmatrix} P(s) & -Q(s) \\ 0 & -A(s) \end{bmatrix},$$

for $M_1(s) = \begin{bmatrix} 1 & h \end{bmatrix}$ and $M_2(s) = 1$, which, in view of Remark 2, shows that Σ implements $\mathcal{C}$.

Now, suppose that we have two types of leading vehicles, E_1 and E_2, whose dynamics are given by

$$E_i : \begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} & -1 \\ 0 & \frac{\mathrm{d}}{\mathrm{d}t}+c_i \end{bmatrix} u = \begin{bmatrix} 0 \\ 1 \end{bmatrix} l, \quad i \in \{1,2\}, \tag{16}$$

where $c_1 = 0.25$ and $c_2 = 0.5$, and l is a latent variable. Note that $\mathfrak{B}_i(E_1) \subset \mathfrak{B}_i(A)$ and $\mathfrak{B}_i(E_2) \subset \mathfrak{B}_i(A)$, hence E_1 and E_2 are environments compatible with $\mathcal{C}$. Consequently, we expect tracking of the headway spacing policy regardless of whether the controlled vehicle Σ is following E_1 or E_2. This is indeed the case, as can be seen in Figure 5.
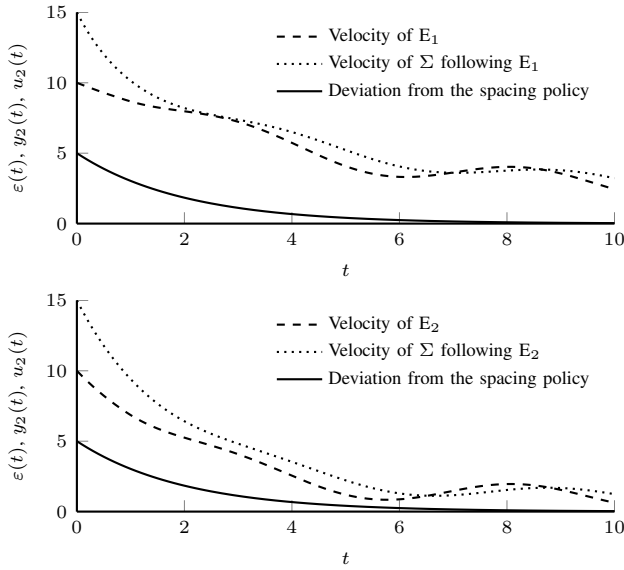
Fig. 5. For each $i \in \{1, 2\}$, the velocity u_2 of the leading vehicle E_i, the velocity y_2 of the controlled vehicle Σ following E_i, and the deviation from the headway spacing policy $\varepsilon = y_1 - u_1 + h y_2$, with l in (16) given by $l(t) = 1$ if $t \in [0, 1]$ and $l(t) = \frac{1}{2} + \sin\left(\frac{3\pi}{8}t + \frac{3\pi}{16}\right)$ if $t > 1$.

To conclude this section, note that the output of the controlled vehicle Σ represents its position and velocity, hence Σ implements the contract $\mathcal{C}' = (A, \Gamma')$ with guarantees

$$\Gamma' : \begin{bmatrix} \frac{d}{dt} & -1 \end{bmatrix} y = 0.$$

Furthermore, Σ is easily seen to implement the conjunction $\mathcal{C} \wedge \mathcal{C}' = (A, \Gamma \wedge \Gamma')$, which we obtain by using Theorem 6. Since $\mathfrak{B}_o(\Gamma \wedge \Gamma') \subset \mathfrak{B}_o(\Gamma') = \mathfrak{B}_i(A)$, an implementation Σ of the conjunction $\mathcal{C} \wedge \mathcal{C}'$ not only tracks the headway spacing policy, but it also guarantees that the interconnection $E \wedge \Sigma$, where E is an environment compatible with $\mathcal{C} \wedge \mathcal{C}'$, can be interpreted as a compatible environment of $\mathcal{C} \wedge \mathcal{C}'$. In other words, by defining appropriate contracts, we can ensure that several interconnected vehicles simultaneously track a given constant headway spacing policy.

VI. CONCLUSION

We presented assume-guarantee contracts for linear dynamical systems. These contracts consist of assumptions and guarantees, both of which are linear systems themselves, and which define a class of compatible environments and a class of implementations, respectively. We then characterized contract implementation on the basis of the assumptions and guarantees alone. This allowed us to derive a necessary condition for contract consistency, which we used to characterize contract refinement. Finally, through the notion of refinement, we also defined contract conjunction and characterized it in two special cases. Future work will focus on extending the theory to include appropriate notions of contract composition, which is crucial in enabling the modular design and analysis of interconnected systems.

APPENDIX

Proof of Theorem 4: Suppose that $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$ and $\mathfrak{B}(A_2 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_2)$. Clearly, the inclusion $\mathfrak{B}_i(A_2) \subset$

$\mathfrak{B}_i(A_1)$ implies that every environment compatible with $\mathcal{C}_2$ is also compatible with $\mathcal{C}_1$, such that the first condition in Definition 5 holds. To show that the second condition also holds, let Σ be an implementation of $\mathcal{C}_1$. This implies that $\mathfrak{B}(A_2 \wedge \Sigma) \subset \mathfrak{B}(\Gamma_1)$ since A_2 is compatible with $\mathcal{C}_1$. But $\mathfrak{B}(A_2 \wedge \Sigma) \subset \mathfrak{B}(\Gamma_1)$ only if $\mathfrak{B}(A_2 \wedge \Sigma) \subset \mathfrak{B}(A_2 \wedge \Gamma_1)$, and $\mathfrak{B}(A_2 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_2)$, hence $\mathfrak{B}(A_2 \wedge \Sigma) \subset \mathfrak{B}(\Gamma_2)$. From Theorem 2 it follows that Σ implements $\mathcal{C}_2$, which shows that every implementation of $\mathcal{C}_1$ is also an implementation of $\mathcal{C}_2$, i.e., the second condition in Definition 5 holds as well.

Conversely, suppose that $\mathcal{C}_1$ refines $\mathcal{C}_2$. Since A_2 is an environment compatible with $\mathcal{C}_2$, it also needs to be compatible with $\mathcal{C}_1$, hence $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$. The only thing left to show is that $\mathfrak{B}(A_2 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_2)$. In view of Theorem 2 and the assumption that $\mathcal{C}_1$ is consistent, there exists a system Σ as in (3) in input-output form such that $\mathfrak{B}(A_1 \wedge \Sigma) \subset \mathfrak{B}(\Gamma_1)$. Note that the latter holds if and only if $\mathfrak{B}(A_1 \wedge \Sigma) \subset \mathfrak{B}(A_1 \wedge \Gamma_1)$. Furthermore, due to Lemma 3, we know there there exist guarantees Γ'_1 of the form (9) with $G'(s)$ that has full row rank such that $\mathfrak{B}(A_1 \wedge \Gamma_1) = \mathfrak{B}(A_1 \wedge \Gamma'_1)$, from which it follows that

$$\mathfrak{B}(A_1 \wedge \Sigma) \subset \mathfrak{B}(A_1 \wedge \Gamma'_1) \subset \mathfrak{B}(\Gamma'_1).$$

Due to Theorem 1, the latter holds if and only if there exist polynomial matrices $M_1(s)$ and $M_2(s)$ satisfying

$$\begin{bmatrix} G'_1(s) & -H'_1(s) \end{bmatrix} = \begin{bmatrix} M_1(s) & M_2(s) \end{bmatrix} \begin{bmatrix} P(s) & -Q(s) \\ 0 & -A_1(s) \end{bmatrix}$$

Note that $M_1(s)$ has full row rank because $G'_1(s)$ has full row rank and $P(s)$ is invertible. This implies that there exists a polynomial matrix $M'_1(s)$ for which $\begin{bmatrix} M_1(s)^\top & M'_1(s)^\top \end{bmatrix}$ is invertible. Let k be a positive integer and define

$$P_k(s) = \begin{bmatrix} M_1(s) \\ s^k M'_1(s) \end{bmatrix} P(s), \quad Q_k(s) = \begin{bmatrix} M_1(s) \\ s^k M'_1(s) \end{bmatrix} Q(s).$$

We will show that the system

$$\Sigma_k : P_k\left(\tfrac{d}{dt}\right) y = Q_k\left(\tfrac{d}{dt}\right) u$$

implements $\mathcal{C}_1$ for any positive integer k. Note that $P_k(s)$ is invertible, $P_k(s)^{-1} Q_k(s) = P(s)^{-1} Q(s)$ is proper, and

$$\begin{bmatrix} G'_1(s) & -H'_1(s) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} & M_2(s) \end{bmatrix} \begin{bmatrix} P_k(s) & -Q_k(s) \\ 0 & -A_1(s) \end{bmatrix},$$

that is, Σ_k is in input-output form and $\mathfrak{B}(A_1 \wedge \Sigma_k) \subset \mathfrak{B}(\Gamma'_1)$. As the latter holds only if $\mathfrak{B}(A_1 \wedge \Sigma_k) \subset \mathfrak{B}(A_1 \wedge \Gamma'_1)$, and $\mathfrak{B}(A_1 \wedge \Gamma'_1) = \mathfrak{B}(A_1 \wedge \Gamma_1)$, it follows that

$$\mathfrak{B}(A_1 \wedge \Sigma_k) \subset \mathfrak{B}(A_1 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_1),$$

hence Σ_k implements $\mathcal{C}_1$ due to Theorem 2. But then Σ_k must also implement $\mathcal{C}_2$, which is the case if and only if $\mathfrak{B}(A_2 \wedge \Sigma_k) \subset \mathfrak{B}(\Gamma_2)$. The latter holds if and only if there exist polynomial matrices $N_{1,k}(s)$ and $N_{2,k}(s)$ such that

$$\begin{bmatrix} G_2(s) & -H_2(s) \end{bmatrix} = \begin{bmatrix} N_{1,k}(s) & N_{2,k}(s) \end{bmatrix} \begin{bmatrix} P_k(s) & -Q_k(s) \\ 0 & -A_2(s) \end{bmatrix}.$$

In particular, we have that

$$G_2(s) = N_{1,k}(s)P_k(s) = N_{1,k}(s)\begin{bmatrix} M_1(s) \\ s^k M_1'(s) \end{bmatrix} P(s).$$

which implies that

$$G_2(s)P(s)^{-1}\begin{bmatrix} M_1(s) \\ M_1'(s) \end{bmatrix}^{-1} = N_{1,k}(s)\begin{bmatrix} I & 0 \\ 0 & s^k I \end{bmatrix}, \quad (17)$$

As the left hand side of (17) is independent of k, we must have that the right-hand side is independent as well. Therefore, if we partition $N_{1,k}(s) = \begin{bmatrix} N_{11,k}(s) & N_{12,k}(s) \end{bmatrix}$, then there exist polynomial matrices $N_{11}(s)$ and $N_{22}(s)$ such that $N_{11,k}(s) = N_{11}(s)$ and $s^k N_{12,k}(s) = N_{12}(s)$ for all positive integers k. This is possible only if $N_{12,k}(s) = 0$ for all positive integers k, which implies that

$$G_2(s) = N_{11}(s)M_1(s)P(s) = N_{11}(s)G_1'(s),$$

and, similarly, that

$$\begin{aligned} H_2(s) &= N_{11}(s)M_1(s)Q(s) + N_{2,k}(s)A_2(s) \\ &= N_{11}(s)(H_1'(s) - M_2(s)A_1(s)) + N_{2,k}(s)A_2(s). \end{aligned}$$

Since $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A_1)$, there exists a polynomial matrix $R(s)$ such that $A_1(s) = R(s)A_2(s)$, and thus

$$\begin{bmatrix} G_2(s) & -H_2(s) \end{bmatrix} = \begin{bmatrix} N_{11}(s) & N_2(s) \end{bmatrix} \begin{bmatrix} G_1'(s) & -H_1'(s) \\ 0 & -A_2(s) \end{bmatrix},$$

where $N_2(s) = N_{11}(s)M_2(s)R(s) - N_{2,k}(s)$ for an arbitrary positive integer k. Finally, this yields $\mathfrak{B}(A_2 \wedge \Gamma_1') \subset \mathfrak{B}(\Gamma_2)$, and since $\mathfrak{B}(A_2 \wedge \Gamma_1') = \mathfrak{B}(A_2 \wedge \Gamma_1)$, we conclude that $\mathfrak{B}(A_2 \wedge \Gamma_1) \subset \mathfrak{B}(\Gamma_2)$, as desired. ∎

Proof of Theorem 6: Due to Lemma 5, we already know that the contract $\mathcal{C}_1 \wedge \mathcal{C}_2$ given in (13) refines both $\mathcal{C}_1$ and $\mathcal{C}_2$. Therefore, we only need to show that $\mathcal{C}_1 \wedge \mathcal{C}_2$ is the largest such contract. To this end, let $\mathcal{C} = (A, \Gamma)$ be a contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$. Then every environment compatible with $\mathcal{C}_1$ or $\mathcal{C}_2$ must also be compatible with $\mathcal{C}$. This is the case if and only if $\mathfrak{B}_i(A_1) \subset \mathfrak{B}_i(A)$ and $\mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A)$, which is equivalent to $\mathfrak{B}_i(A_1) + \mathfrak{B}_i(A_2) \subset \mathfrak{B}_i(A)$ because $\mathfrak{B}_i(A)$ is a linear space. The latter can be written as $\mathfrak{B}_i(A_1 \vee A_2) \subset \mathfrak{B}_i(A)$, which implies that every environment compatible with $\mathcal{C}_1 \wedge \mathcal{C}_2$ is also compatible with $\mathcal{C}$.

Next, we will show that every implementation of $\mathcal{C}$ is also an implementation of $\mathcal{C}_1 \wedge \mathcal{C}_2$. This is vacuously true when $\mathcal{C}$ is not consistent. Suppose that $\mathcal{C}$ is consistent. Since $\mathcal{C}$ refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, from Theorem 2 it follows that

$$\mathfrak{B}(A_1 \wedge \Gamma) \subset \mathfrak{B}(\Gamma_1) \quad \text{and} \quad \mathfrak{B}(A_2 \wedge \Gamma) \subset \mathfrak{B}(\Gamma_2). \quad (18)$$

Furthermore, since $\mathfrak{B}_i(A_1 \vee A_2) \subset \mathfrak{B}_i(A)$, the same theorem tells us that $\mathcal{C}$ refines $\mathcal{C}_1 \wedge \mathcal{C}_2$ if and only if

$$\mathfrak{B}((A_1 \vee A_2) \wedge \Gamma) \subset \mathfrak{B}(\Gamma_1 \wedge \Gamma_2). \quad (19)$$

With this in mind, we will show that (18) implies (19) in the following two cases.

1) Suppose that $\mathfrak{B}_i(A_1) = \mathfrak{B}_i(A_2)$. Then

$$\mathfrak{B}_i(A_1 \vee A_2) = \mathfrak{B}_i(A_1) = \mathfrak{B}_i(A_2), \quad (20)$$

from which it follows that

$$\mathfrak{B}(A_1 \wedge \Gamma) = \mathfrak{B}(A_2 \wedge \Gamma) = \mathfrak{B}((A_1 \vee A_2) \wedge \Gamma),$$

and thus (18) implies (19).

2) Suppose that

$$\mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_1) = \mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_2) \quad (21)$$

Let $i \in \{1, 2\}$ and note that (18) holds only if

$$\mathfrak{B}(A_i \wedge \Gamma) \subset \mathfrak{B}(A_i \wedge \Gamma_i). \quad (22)$$

Since $\mathfrak{B}_i(A_i) \subset \mathfrak{B}_i(A_1 \vee A_2)$, (22) yields

$$\mathfrak{B}(A_i \wedge \Gamma) \subset \mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_i). \quad (23)$$

Furthermore, from (21) it follows that

$$\mathfrak{B}((A_1 \vee A_2) \wedge \Gamma_i) \subset \mathfrak{B}(\Gamma_1 \wedge \Gamma_2), \quad (24)$$

which, together with (23), implies that

$$\mathfrak{B}(A_i \wedge \Gamma) \subset \mathfrak{B}(\Gamma_1 \wedge \Gamma_2) \quad (25)$$

But $\mathfrak{B}(\Gamma_1 \wedge \Gamma_2)$ is a linear space, hence

$$\mathfrak{B}(A_1 \wedge \Gamma) + \mathfrak{B}(A_2 \wedge \Gamma) \subset \mathfrak{B}(\Gamma_1 \wedge \Gamma_2). \quad (26)$$

Now, note that (19) would follow from (26) if

$$\mathfrak{B}((A_1 \vee A_2) \wedge \Gamma) \subset \mathfrak{B}(A_1 \wedge \Gamma) + \mathfrak{B}(A_2 \wedge \Gamma). \quad (27)$$

To show that (27) holds, let $(u, y) \in \mathfrak{B}((A_1 \vee A_2) \wedge \Gamma)$. It follows that $u \in \mathfrak{B}_i(A_1 \vee A_2)$ and $(u, y) \in \mathfrak{B}(\Gamma)$, which implies that $u = u_1 + u_2$, where $u_1 \in \mathfrak{B}_i(A_1)$ and $u_2 \in \mathfrak{B}_i(A_2)$. Since $\mathcal{C}$ is consistent, we must have that $\mathfrak{B}_i(A) \subset \mathfrak{B}_i(\Gamma)$. Therefore, for all $u \in \mathfrak{B}_i(A)$, there exists y such that $(u, y) \in \mathfrak{B}_i(\Gamma)$. Given that $\mathfrak{B}_i(A_1) \subset \mathfrak{B}_i(A)$, this means that there exist y_1 such that $(u_1, y_1) \in \mathfrak{B}(\Gamma)$. Since $\mathfrak{B}(\Gamma)$ is a linear space and $(u, y) \in \mathfrak{B}(\Gamma)$, we get $(u_2, y - y_2) \in \mathfrak{B}(\Gamma)$. But then $(u_1, y_1) \in \mathfrak{B}(A_1 \wedge \Gamma)$ and $(u_2, y - y_1) \in \mathfrak{B}(A_2 \wedge \Gamma)$, hence $(u, y) \in \mathfrak{B}(A_1 \wedge \Gamma) + \mathfrak{B}(A_2 \wedge \Gamma)$, which shows that (27) holds, as desired.

In both cases, we have seen that (19) holds, hence $\mathcal{C}$ refines $\mathcal{C}_1 \wedge \mathcal{C}_2$. Since this is the case for any $\mathcal{C}$ that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$, we conclude that the contract $\mathcal{C}_1 \wedge \mathcal{C}_2$ given in (13) is indeed the largest contract that refines both $\mathcal{C}_1$ and $\mathcal{C}_2$. ∎

REFERENCES

[1] B. Meyer, "Applying 'design by contract'," *Computer*, vol. 25, no. 10, pp. 40–51, 1992.

[2] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. A. Henzinger, and K. G. Larsen, *Contracts for System Design*. Foundations and Trends in Electronic Design Automation, Now Publishers, 2018.

[3] J. C. Willems, "Dissipative dynamical systems part I: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[4] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[5] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer US, 2009.

[6] B. M. Shali, B. Besselink, and A. J. van der Schaft, "Behavioural contracts for linear dynamical systems: input assumptions and output guarantees," in *Proceedings of the European Control Conference*, pp. 564–569, 2021.

[7] B. Besselink, K. H. Johansson, and A. van der Schaft, "Contracts as specifications for dynamical systems in driving variable form," in *Proceedings of the European Control Conference*, pp. 263–268, 2019.

[8] E. S. Kim, M. Arcak, and S. A. Seshia, "A small gain theorem for parametric assume-guarantee contracts," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pp. 207–216, 2017.

[9] M. Al Khatib and M. Zamani, "Controller synthesis for interconnected systems using parametric assume-guarantee contracts," in *Proceedings of the American Control Conference*, pp. 5419–5424, 2020.

[10] Y. Chen, J. Anderson, K. Kalsi, A. D. Ames, and S. H. Low, "Safety-critical control synthesis for network systems with control barrier functions and assume-guarantee contracts," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 487–499, 2021.

[11] A. Saoud, A. Girard, and L. Fribourg, "On the composition of discrete and continuous-time assume-guarantee contracts for invariance," in *Proceedings of the European Control Conference*, pp. 435–440, 2018.

[12] A. Saoud, A. Girard, and L. Fribourg, "Contract based design of symbolic controllers for interconnected multiperiodic sampled-data systems," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 773–779, 2018.

[13] A. Saoud, A. Girard, and L. Fribourg, "Contract-based design of symbolic controllers for safety in distributed multiperiodic sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1055–1070, 2021.

[14] I. D. Loreto, A. Borri, and M. D. Di Benedetto, "An assume-guarantee approach to sampled-data quantized glucose control," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 3401–3406, 2020.

[15] D. Zonetti, A. Saoud, A. Girard, and L. Fribourg, "A symbolic approach to voltage stability and power sharing in time-varying dc microgrids," in *Proceedings of the European Control Conference*, pp. 903–909, 2019.

[16] A. Eqtami and A. Girard, "A quantitative approach on assume-guarantee contracts for safety of interconnected systems," in *Proceedings of the European Control Conference*, pp. 536–541, 2019.

[17] K. Ghasemi, S. Sadraddini, and C. Belta, "Compositional synthesis via a convex parameterization of assume-guarantee contracts," in *Proceedings of the International Conference on Hybrid Systems: Computation and Control*, pp. 1–10, 2020.

[18] M. Sharf, B. Besselink, A. Molin, Q. Zhao, and K. Henrik Johansson, "Assume/guarantee contracts for dynamical systems: Theory and computational tools," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 25–30, 2021.

[19] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*. Springer-Verlag New York, 1998.

[20] J. C. Willems, "The behavioral approach to open and interconnected systems," *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, 2007.

[21] J. C. Willems, "Input-output and state-space representations of finite-dimensional linear time-invariant systems," *Linear Algebra and its Applications*, vol. 50, pp. 581–608, 1983.

[22] J. W. Polderman, "A new and simple proof of the equivalence theorem for behaviors," *Systems & Control Letters*, vol. 41, no. 3, pp. 223–224, 2000.