Implementation of differential-algebraic equations contracts

Yahao Chen[1], Albert Benveniste[1], Benoît Caillaud[1]

Abstract— **In this paper, we extend several notions from contract theory in computer science to dynamical systems. In particular, we focus on systems described by linear differential-algebraic equations (DAEs). More specifically, we formulate assume-guarantee contracts in which all components are represented in a unified linear DAE form. To compare the behaviors of DAEs, we employ the simulation relations for DAEs proposed in [9]. We then investigate how to implement DAE contracts under these simulation relations. Linear systems geometric theory is used to characterize necessary and sufficient conditions for the existence of an implementation. Based on these conditions, we develop an algorithm that constructs an implementation step by step. We discuss the differences and connections between the proposed DAE-based contracts and contracts for input-output systems studied in the existing literature. Finally, an electrical circuit example is given to show the implementation of DAE contract.**

I. INTRODUCTION

Modern engineering systems, such as automotive and aerospace platforms, power grids, and cyber-physical systems, are built by interconnecting many components that are designed, and often supplied, by different teams. Verifying such systems monolithically scales poorly, which has motivated *compositional* design methodologies, in which each component is specified, implemented, and verified separately, and system-level properties follow from component-level ones. *Contract theory* is one of the most successful frameworks for this purpose [1], [11]. A particular case is *assume-guarantee (A/G) contract* for system design [1]: a contract is a pair of assumptions on the environment and guarantees offered by the component when the assumptions hold. Contracts are equipped with an algebra of operations, namely refinement, composition, conjunction, and quotient, that supports independent implementation, component replacement, and top-down requirement decomposition [10].

a) Contracts for dynamical systems.: Contract theory was developed mostly for discrete and reactive systems, whose behaviors are sets of traces. Carrying it over to dynamical systems, where components are described by differential or difference equations, raises the question of what assumptions, guarantees, and implementation should mean for continuous-time models. Several works address this. Parametric contracts, which express input-output gain properties, were studied in [17] for discrete-time systems, with computational tools based on linear programming [16]. For continuous-time linear systems, contracts capturing set-invariance properties on input and output spaces were proposed in [12]. Besselink, Johansson, and van der Schaft [4] defined contracts in driving variable form, in which assumptions and guarantees are themselves semi-explicitly linear inputs-outputs DAEs compared through simulation relations, and this was further developed for contracts implementation in [14]. Closer to the behavioral viewpoint, [13] defined contracts for linear systems with inputs and outputs whose assumptions and guarantees are characterized as behaviors, and derived necessary conditions for the existence of an implementation. These works model components as state-space systems with a prescribed input-output structure, or as driving-variable systems, and treat composition mainly in cascade form [16], [13], where the output of one component feeds the input of the next. Bidirectional couplings, as in physical networks, must first be recast into this structure.

b) Differential-algebraic equations.: Differential-algebraic equations (DAEs) are the natural modeling language for interconnected physical systems. Network models in electrical, mechanical, and thermo-fluid engineering, and the equation-based modeling languages built on them (e.g. Modelica), produce DAEs because interconnection constraints, such as Kirchhoff's laws or rigid couplings, are algebraic equations [6], [7]. Interconnecting ODEs through such constraints therefore yields a DAE, and interconnecting DAEs yields again a DAE, so the class is closed under interconnection, which ODEs are not. DAEs also fit the behavioral approach to systems theory [21], in which a system is a set of trajectories of external variables and no a priori partition into inputs, outputs, and states is imposed. This is precisely the setting of contracts, where the assumptions and guarantees are sets of behaviors. A further feature is that DAEs admit inconsistent initial conditions, which, together with the generalized (distributional) solution theory [18], allows impulsive behavior, and such inconsistency can arise precisely from interconnecting components. Standard input-output systems are themselves DAEs once u and y are regarded as algebraic variables. Behavioral comparison of DAEs is provided by the simulation and bisimulation relations of [9], which extend the corresponding notions for linear systems [20] to the DAE setting. Geometric methods based on controlled and conditioned invariant subspaces [19], and their extension to singular systems [7], provide the tools for characterizing such relations.

To the best of our knowledge, no contract framework has been developed in which all components, assumptions, and guarantees are uniformly linear DAEs. Such a setting

*This work was supported by ANR project GFdDAE:ANR-25-CE48-4916 and Rennes Metropole l'allocation d'installation Scientifique.

[1] Yahao Chen, Albert Benveniste and Benoît Caillaud with INRIA center at Rennes University; yahao.chen@inria.fr, albert.benveniste@inria.fr, benoit.caillaud@inria.fr

offers several advantages over the existing ones. First, every object in the framework (the component, the assumption, the guarantee, and the interconnected system) is a DAE, so one representation serves all roles and no conversion between model classes is needed. Second, interconnections are expressed directly as algebraic constraints on the variables being connected, so composition does not require an input-output partition. Third, simulation relations remain available for composed systems, so contract reasoning can be carried out on the interconnection itself.

This paper is organized as follows. Section II presents the preliminaries on DAE simulations. Section III discusses the composition of DAEs and their properties. In Section IV, we establish necessary and sufficient conditions for the existence of implementations of DAE contracts and develop an algorithm for constructing such implementations. Section V illustrates the proposed approach through a circuit example. Finally, Section VI presents the conclusions.

II. Preliminaries on simulations of DAEs

Consider linear DAEs of the form

$$\Delta: \quad E\dot{x} = Hx, \quad w = Lx, \tag{1}$$

where $x \in \mathcal{X} = \mathbb{R}^n$ are generalized state variables, $E : \mathcal{X} \to \mathbb{R}^l$, $H : \mathcal{X} \to \mathbb{R}^l$ and $L : \mathcal{X} \to \mathbb{R}^m$ are linear maps. A $\mathcal{C}^1$-curve $x : \mathbb{R} \to \mathcal{X}$ is called a *solution* of a DAE Δ if $\forall t \geq 0 : E\dot{x}(t) = Hx(t)$. We say a subspace $\mathcal{V}$ is (H, E)-invariant if $H\mathcal{V} \subseteq E\mathcal{V}$. Define the subspace $\mathcal{V}^* \subseteq \mathcal{X}$ as the *largest* (H, E)*-invariant*. We will also call $\mathcal{V}^*$ the *consistency space* of Δ. It is well-known [2], [5] that for any $x_0 \in \mathcal{V}^*$, there exists a solution $x(t)$ starting from $x(0) = x_0$ and $\forall t > 0 : x(t) \in \mathcal{V}^*$. The consistency space can be constructed via $\mathcal{V}^* = \mathcal{V}^n$, where $\mathcal{V}^n$ is the limit of the Wong sequence [22]:

$$\mathcal{V}^0 = \mathcal{X}, \quad \mathcal{V}^{j+1} = H^{-1}E\mathcal{V}^j, \ j \in \mathbb{N}. \tag{2}$$

The variables w are called the *shared variables* of the DAEs. To simplify the design procedure, we assume that

$$\operatorname{rank} \begin{bmatrix} E \\ L \end{bmatrix} = n. \tag{3}$$

We consider DAE components with common shared variables:

$$\Delta_i: \quad E_i\dot{x} = H_i x, \quad w = L_i x.$$

Two DAEs $\Delta_1 = (E_1, H_1, L_1)$ and $\Delta_2 = (E_2, H_2, L_2)$ are called *externally equivalent* [5], or shortly, *ex-equivalent*, if there exist two invertible matrices Q and P such that $E_2 = QE_1P^{-1}$, $H_2 = QH_1P^{-1}$, $L_2 = L_1P^{-1}$. It is clearly that ex-equivalence preserves solutions, i.e., $x(t)$ is a solution of Δ_1 if and only if $Px(t)$ is a solution of Δ_2.

Remark 1. (i) A given DAE Δ is always ex-equivalent to a semi-explicit DAE $\tilde{\Delta}$: $\left[\begin{smallmatrix} I & 0 \\ 0 & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} \dot{x}_1 \\ \dot{x}_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} H_1 & H_2 \\ H_3 & H_4 \end{smallmatrix}\right] \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right]$. Hence the shared variables in the new coordinated DAE $\tilde{\Delta}$ and satisfying assumption (3) can be $w = x_2$, which are all algebraic variables.

(ii) Note that we can always extract differential variables Cx of Δ as shared variables by adding the algebraic equations $y = Cx$ to Δ. Then the extended system becomes $\left[\begin{smallmatrix} E & 0 \\ 0 & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} \dot{x} \\ \dot{y} \end{smallmatrix}\right] = \left[\begin{smallmatrix} H & 0 \\ -C & I \end{smallmatrix}\right] \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ and y becomes algebraic variables of the extended system.

To connect two DAEs, we define the composition of DAEs as follows.

Definition 1 (Composition of DAEs)**.** Given two DAEs $\Delta_1 = (E_1, H_1, L_1)$ and $\Delta_2 = (E_2, H_2, L_2)$, the composition of the two DAEs is $\Delta_1 \times \Delta_2 := (E_{12}, H_{12}, L_{12})$, where

$$E_{12} := \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \\ 0 & 0 \end{bmatrix}, \quad H_{12} := \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \\ L_1 & -L_2 \end{bmatrix}, \quad L_{12} = \begin{bmatrix} L_1 & 0 \end{bmatrix}.$$

Note that the matrix L_{12} can be also chosen as $L_{12} = \left[\begin{smallmatrix} 0 & L_2 \end{smallmatrix}\right]$ because of the constraints $w = L_1x_1 = L_2x_2$. With the help of the notion of compositions, to compare the behaviors of two DAEs, we reform the simulation relations of DAE introduced in [9], [15] as follows.

Definition 2 (Simulation relations)**.** Given two DAEs $\Delta_1 = (E_1, H_1, L_1)$, $\Delta_2 = (E_2, H_2, L_2)$ and their composition $\Delta_1 \times \Delta_2 = (E_{12}, H_{12}, L_{12})$. A subspace $\mathcal{S} \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is called a *simulation relation* of Δ_1 by Δ_2 if

(i) $\mathcal{S}$ is (H_{12}, E_{12})-invariant;

(ii) for every $(x_{10}, x_{20}) \in \mathcal{S}$ and every solution $x_1(\cdot)$ of Δ_1 with $x_1(0) = x_{10}$, there exists a solution $x_2(\cdot)$ of Δ_2 with $x_2(0) = x_{20}$ such that $(x_1(\cdot), x_2(\cdot))$ is a solution of $\Delta_1 \times \Delta_2$.

If, additionally, $\pi_{\mathcal{X}_1}(\mathcal{S}) = \mathcal{V}_1^*$, then we call $\mathcal{S}$ a *full simulation relation*.

Proposition 1 ([9])**.** *A subspace $\mathcal{S} \subseteq \mathcal{X}_1 \times \mathcal{X}_2$ is a simulation relation of Δ_1 by Δ_2 if and only if*

$$\begin{bmatrix} H_1 & 0 \\ 0 & H_2 \\ L_1 & -L_2 \end{bmatrix} \mathcal{S} \subseteq \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \\ 0 & 0 \end{bmatrix} \mathcal{S}, \tag{4}$$

$$\begin{bmatrix} \ker E_1 \cap \mathcal{V}_1^* \\ 0 \end{bmatrix} \subseteq \mathcal{S} + \begin{bmatrix} 0 \\ \ker E_2 \cap \mathcal{V}_2^* \end{bmatrix}. \tag{5}$$

Definition 3. A DAE Δ_2 is said to be *simulated by* Δ_1, denoted by $\Delta_1 \preccurlyeq \Delta_2$, if there exists a simulation relation $\mathcal{S}$ of Δ_1 by Δ_2. We call two DAEs Δ_1 and Δ_2 are *equivalent in the sense of simulation*, or simply *s-equivalent*, denoted by $\Delta_1 \overset{\bullet}{=} \Delta_2$ if $\Delta_1 \preccurlyeq \Delta_2$ and $\Delta_2 \preccurlyeq \Delta_1$.

Throughout the paper, the terms "larger" and "smaller" when used to compare DAEs, are understood with respect to the simulation relation.

Remark 2. With the help of Proposition 1, it can be shown that if two DAEs are ex-equivalent, then they are also s-equivalent. The converse, however, does not hold in general.

III. Properties of simulations and compositions

In this section, we study some properties regarding simulation relations and compositions of DAEs.

Proposition 2 (Commutativity and associativity)**.** *Consider DAEs Δ_1, Δ_2, Δ_3, then*

(i) $\Delta_1 \times \Delta_2 \overset{\bullet}{=} \Delta_2 \times \Delta_1$.

(ii) $(\Delta_1 \times \Delta_2) \times \Delta_3 \overset{\bullet}{=} \Delta_1 \times (\Delta_2 \times \Delta_3)$.

Proof. (i) It is clear that $\Delta_1 \times \Delta_2$ and $\Delta_2 \times \Delta_1$ are ex-equivalent via $Q = I$ and a permutation matrix $P = \left[\begin{smallmatrix} 0 & I_{n_2} \\ I_{n_1} & 0 \end{smallmatrix}\right]$, they are thus s-equivalent by Remark 2.

(ii) By definition, we have

$$(\Delta_1 \times \Delta_2) \times \Delta_3 : \begin{bmatrix} E_1 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & E_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ L_1 & -L_2 & 0 \\ 0 & 0 & H_3 \\ 0 & L_2 & -L_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

and

$$\Delta_1 \times (\Delta_2 \times \Delta_3) : \begin{bmatrix} E_1 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ 0 & 0 & H_3 \\ 0 & L_2 & -L_3 \\ L_1 & -L_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Moreover, we have $L_{12,3} = L_{1,23}$ and they can be $\left[\begin{smallmatrix} L_1 & 0 & 0 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 0 & L_2 & 0 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} 0 & 0 & L_3 \end{smallmatrix}\right]$. Clearly, $(\Delta_1 \times \Delta_2) \times \Delta_3$ and $\Delta_1 \times (\Delta_2 \times \Delta_3)$ are ex-equivalent with a permutation matrix Q and $P = I$. Hence they are also s-equivalent. □

Proposition 3. *Given DAEs Δ_1, Δ_2 and Δ, the following results hold:*

(i) *If $\Delta_1 \preccurlyeq \Delta_2$, then $\Delta_1 \times \Delta \preccurlyeq \Delta_2 \times \Delta$.*

(ii) *If $\Delta_1' \preccurlyeq \Delta_1$ and $\Delta_2' \preccurlyeq \Delta_2$, then $\Delta_1' \times \Delta_2' \preccurlyeq \Delta_1 \times \Delta_2$.*

(iii) $\Delta_1 \times \Delta \preccurlyeq \Delta_2 \Leftrightarrow \forall \Delta' \preccurlyeq \Delta : \Delta_1 \times \Delta' \preccurlyeq \Delta_2$.

Moreover, the above relations still hold when $\preccurlyeq$ represents full simulation relations.

Proof. (i) If $\Delta_1 \preccurlyeq \Delta_2$, then there exists a simulation relation $\mathcal{S}$ of Δ_1 by Δ_2. Consider the DAEs $\Delta_1 \times \Delta$ and $\Delta_2 \times \Delta$ and let $\mathcal{V}_{1m}^*$ and $\mathcal{V}_{2m}^*$ be their consistency spaces, respectively. Define

$$\mathcal{S}' = \{(x_1, y, x_2, y) \mid \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] \in \mathcal{S}, \left[\begin{smallmatrix} x_1 \\ y \end{smallmatrix}\right] \in \mathcal{V}_{1m}^*, \left[\begin{smallmatrix} x_2 \\ y \end{smallmatrix}\right] \in \mathcal{V}_{2m}^* \}.$$

We intend to prove that $\mathcal{S}'$ is a simulation relation of $\Delta_1 \times \Delta$ by $\Delta_2 \times \Delta$. By the linearity of $\mathcal{S}$, $\mathcal{V}_{1m}^*$ and $\mathcal{V}_{2m}^*$, it is clear that $\mathcal{S}'$ is also a linear subspace. Then by $\left[\begin{smallmatrix} L_1 & -L_2 \end{smallmatrix}\right] \mathcal{S} = \{0\}$ of (4) and the definition of $\mathcal{S}'$, we can deduce $L_1 x_1 = L_2 x_2 = Ly$ for any $(x_1, y, x_2, y) \in \mathcal{S}'$, thus

$$\begin{bmatrix} H_1 & 0 & 0 & 0 \\ 0 & H & 0 & 0 \\ L_1 & -L & 0 & 0 \\ 0 & 0 & H_2 & 0 \\ 0 & 0 & 0 & H \\ 0 & 0 & L_2 & -L \\ 0 & L & 0 & -L \end{bmatrix} \mathcal{S}' \subseteq \begin{bmatrix} E_1 & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & E_2 & 0 \\ 0 & 0 & 0 & E \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathcal{S}'.$$

Notice that by (5), for any $e_1 \in \ker E_1 \cap \mathcal{V}_1^*$, there exist $(x_1, x_2) \in \mathcal{S}$ and $e_2 \in \ker E_2 \cap \mathcal{V}_2^*$ such that $x_1 = e_1$ and $e_2 = -x_2$.

Now consider any $(e_1', e) \in \ker \left[\begin{smallmatrix} E_1 & 0 \\ 0 & E \end{smallmatrix}\right] \cap \mathcal{V}_{1m}^*$. As $\ker \left[\begin{smallmatrix} E_1 & 0 \\ 0 & E \end{smallmatrix}\right] \cap \mathcal{V}_{1m}^* \subseteq \begin{bmatrix} \ker E_1 \cap \mathcal{V}_1^* & 0 \\ 0 & \ker E \cap \mathcal{V}^* \end{bmatrix}$, we have $e_1' \in \ker E_1 \cap \mathcal{V}_1^*$. It implies that there exists x_2' such that $(x_1', x_2') \in \mathcal{S}$ and $e_2' \in \ker E_2 \cap \mathcal{V}_2^*$ such that $x_1' = e_1'$ and $e_2' = -x_2'$. Let $y = e$, it follows that $(x_1', y) = (e_1', e) \in \mathcal{V}_{1m}^*$.

By the definition of $\mathcal{V}_{1m}^*$, starting from $(x_1', y) \in \mathcal{V}_{1m}^*$, there exists a solution $(x_1'(t), y(t))$ such that $\forall t > 0 : E_1 \dot{x}_1'(t) = H_1 x_1'(t), E\dot{y}(t) = Hy(t), L_1 x_1'(t) = Ly(t)$. Because $(x_1', x_2') \in \mathcal{S}$, for any solution $x_1'(t)$ starting from x_1', there exists $x_2'(t)$ with $x_2'(0) = x_2'$ such that $\forall t > 0 : E_1 \dot{x}_1'(t) = H_1 x_1'(t), E_2 \dot{x}_1'(t) = H_2 x_2'(t), L_2 x_2'(t) = L_1 x_1'(t)$. It follows that starting from (x_2', y), there exists a solution $(x_2'(t), y(t))$ such that $\forall t > 0 : E_2 \dot{x}_2'(t) = H_2 x_2'(t), E\dot{y}(t) = Hy(t), L_2 x_2'(t) = Ly(t)$. Thus we can conclude $(x_2', y) \in \mathcal{V}_{2m}^*$.

It can be seen now that for any $(e_1', e) \in \ker \left[\begin{smallmatrix} E_1 & 0 \\ 0 & E \end{smallmatrix}\right] \cap \mathcal{V}_{1m}^*$, there exist $(x_1', y, x_2', y) \in \mathcal{S}'$ and $(e_2', e') \in \ker \left[\begin{smallmatrix} E_2 & 0 \\ 0 & E \end{smallmatrix}\right] \cap \mathcal{V}_{2m}^*$ such that

$$\begin{bmatrix} e_1' \\ e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1' \\ y \\ x_2' \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_2' \\ e' \end{bmatrix},$$

which implies

$$\begin{bmatrix} \ker \left[\begin{smallmatrix} E_1 & 0 \\ 0 & E \\ 0 & \end{smallmatrix}\right] \cap \mathcal{V}_{1m}^* \end{bmatrix} \subseteq \mathcal{S}' + \begin{bmatrix} 0 \\ \left[\begin{smallmatrix} E_2 & 0 \\ 0 & E \end{smallmatrix}\right] \cap \mathcal{V}_{2m}^* \end{bmatrix}.$$

Hence $\mathcal{S}'$ is a simulation relation of $\Delta_1 \times \Delta$ by $\Delta_2 \times \Delta$ and we have $\Delta_1 \times \Delta \preccurlyeq \Delta_2 \times \Delta$ by Proposition 1. Moreover, $\mathcal{S}'$ is a full simulation relation when $\mathcal{S}$ is, because if $\pi_{\mathcal{X}_1}(\mathcal{S}) = \mathcal{V}_1^*$, then $\pi_{\mathcal{X}_1 \times \mathcal{Y}}(\mathcal{S}') = \mathcal{V}_{1m}^*$ by construction.

(ii) By item (i) above, if $\Delta_1' \preccurlyeq \Delta_1$, then $\Delta_1' \times \Delta_2' \preccurlyeq \Delta_1 \times \Delta_2'$. Moreover, if $\Delta_2' \preccurlyeq \Delta_2$, then because of commutativity and item (i), we have $\Delta_1 \times \Delta_2' \overset{\bullet}{=} \Delta_2' \times \Delta_1 \preccurlyeq \Delta_2 \times \Delta_1 \overset{\bullet}{=} \Delta_1 \times \Delta_2$. Hence $\Delta_1' \times \Delta_2' \preccurlyeq \Delta_1 \times \Delta_2$.

(iii)"$\Leftarrow$" is clear by taking $\Delta = \Delta'$. "$\Rightarrow$": By item (i) and commutativity, if $\Delta' \preccurlyeq \Delta$, then $\Delta_1 \times \Delta' \preccurlyeq \Delta_1 \times \Delta$. Hence if $\Delta_1 \times \Delta \preccurlyeq \Delta_2$, then $\Delta_1 \times \Delta' \preccurlyeq \Delta_2$.

It is also clear that $\preccurlyeq$ in items (ii)-(iii) can be also full simulation relations as they are proved as the corollaries of item (i). □

IV. Implementations of DAE contracts

Now we consider two DAEs Δ_a and Δ_g and call them, respectively, assumption:

$$\Delta_a : \quad E_a \dot{x}_a = H_a x_a, \quad w = L_a x_a,$$

and guarantee:

$$\Delta_g : \quad E_g \dot{x}_g = H_g x_g, \quad w = L_g x_g.$$

Definition 4. A *DAE assumption and guarantee contract*, or shortly, a *DAE contract*, is a pair $\mathcal{C} = (\Delta_a, \Delta_g)$. A DAE Δ_e is called an *environment* of $\mathcal{C}$ if $\Delta_e \preccurlyeq \Delta_a$ with a *full* simulation relation. The set of all environments of $\mathcal{C}$ is denoted by $\mathcal{E}_\mathcal{C}$. A DAE $\Delta_m = (E_m, H_m, L_m)$ is called an *implementation* of $\mathcal{C}$ if $\Delta_a \times \Delta_m \preccurlyeq \Delta_g$ with a *full* simulation relation. The set of all implements of $\mathcal{C}$ is denoted by $\mathcal{M}_\mathcal{C}$.

Clearly, the set $\mathcal{E}_\mathcal{C}$ is not empty and the largest environment in $\mathcal{E}_\mathcal{C}$ is $\Delta_e \overset{\bullet}{=} \Delta_a$. Moreover, the set $\mathcal{M}_\mathcal{C}$ is not empty either, there are actually two "obvious" implementations of a contract $\mathcal{C} = (\Delta_a, \Delta_g)$. One is $\Delta_m = \Delta_g$ because $\Delta_a \preccurlyeq \Delta_t$, where Δ_t is a trivial DAE as $0\dot{w} = 0$, and by Proposition 3, $\Delta_a \times \Delta_g \preccurlyeq \Delta_t \times \Delta_g \overset{\bullet}{=} \Delta_g$. Another one is $\Delta_m = \Delta_a \times \Delta_g$ since $\Delta_a \times (\Delta_a \times \Delta_g) \overset{\bullet}{=} (\Delta_a \times \Delta_a) \times \Delta_g \preccurlyeq \Delta_g$ by Propositions 2 and 3. However, both of those implementations are

not necessarily the largest implementation of the contract $\mathcal{C}$. The problem of finding the largest implementation plays an important role in the original meta theory [1] for contracts design in computer science. The largest implementation $\bar{\Delta}_m$ provides an upper bound of the set $\mathcal{M}_{\mathcal{C}}$. If $\bar{\Delta}_m$ can be constructed, then given any DAE Δ_m, we only need to check $\Delta_m \preccurlyeq \bar{\Delta}_m$ to verify if it is an implementation by the results in item(iii) of Proposition 3.

A. Necessary and sufficient condition for the existence of DAE contracts implementation

In this subsection, we provide a geometric characterization for the existence of implementations of DAE contract, which can be seen as a guideline to find the "largest" implementation.

Consider the composition $\Delta_1 \times \Delta_2$ of DAEs in Definition 1. The shared variables w are duplicated in (x_1, x_2)-coordinates. Because of the constraints $L_1 x_1 = L_2 x_2 = w$, we can always compress the duplicated variable w to have an equivalent representation:

$$\Delta_1 \times \Delta_2 \overset{\bullet}{=} \Delta'_{12}.$$

Find matrices $\bar{L}_1$ and $\bar{L}_2$ such that $\begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix}$ and $\begin{bmatrix} L_2 \\ \bar{L}_2 \end{bmatrix}$ are invertible, define $\left[\begin{smallmatrix} z_1 \\ w \end{smallmatrix}\right] = \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix} x_1$ and $\left[\begin{smallmatrix} w \\ z_2 \end{smallmatrix}\right] = \begin{bmatrix} L_2 \\ \bar{L}_2 \end{bmatrix} x_2$ to get $\Delta'_{12} = (E'_{12}, H'_{12}, L'_{12})$:

$$\begin{aligned} \begin{bmatrix} E_1^1 & E_1^2 & 0 \\ 0 & E_2^2 & E_2^1 \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{w} \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} H_1^1 & H_1^2 & 0 \\ 0 & H_2^2 & H_2^1 \end{bmatrix} \begin{bmatrix} z_1 \\ w \\ z_2 \end{bmatrix}, \\ w &= \begin{bmatrix} 0 & I & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ w \\ z_2 \end{bmatrix}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} \left[\begin{smallmatrix} E_1^1 & E_1^2 \end{smallmatrix}\right] &= E_1 \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix}^{-1}, \ \left[\begin{smallmatrix} H_1^1 & H_1^2 \end{smallmatrix}\right] = H_1 \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix}^{-1}, \\ \left[\begin{smallmatrix} E_2^2 & E_2^1 \end{smallmatrix}\right] &= E_2 \begin{bmatrix} L_2 \\ \bar{L}_2 \end{bmatrix}^{-1}, \ \left[\begin{smallmatrix} H_2^2 & H_2^1 \end{smallmatrix}\right] = H_2 \begin{bmatrix} L_2 \\ \bar{L}_2 \end{bmatrix}^{-1}. \end{aligned}$$

Before presenting the main results, we need the following lemma, which can be seen as a modification of Proposition 1 for characterizing simulation relations using DAEs composition in the reduced coordinates.

Lemma 1. *Consider two DAEs $\Delta_1 = (E_1, H_1, L_1)$ and $\Delta_2 = (E_2, H_2, L_2)$. Let $\Delta'_{12} = (E'_{12}, H'_{12}, L'_{12}) \overset{\bullet}{=} \Delta_1 \times \Delta_2$ given by* (6)*. Assume that Δ_1 and Δ_2 satisfy condition* (3)*. Then $\Delta_1 \preccurlyeq \Delta_2$ if and only if there exists a subspace $\mathcal{S}' \subseteq \mathbb{R}^{n_1+n_2-m}$ such that*

$$H'_{12}\mathcal{S}' \subseteq E'_{12}\mathcal{S}' \tag{7}$$

and

$$\ker E'_{12} \cap \mathcal{V}'_1 \subseteq \ker E'_{12} \cap \mathcal{S}', \tag{8}$$

where $\mathcal{V}'_1 = \left\{ (z_1, w, z_2) \,\middle|\, \left[\begin{smallmatrix} z_1 \\ w \end{smallmatrix}\right] \in \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix} \mathcal{V}_1^* \right\}$ and $\mathcal{V}'_2 = \left\{ (z_1, w, z_2) \,\middle|\, \left[\begin{smallmatrix} w \\ z_2 \end{smallmatrix}\right] \in \begin{bmatrix} L_2 \\ \bar{L}_2 \end{bmatrix} \mathcal{V}_2^* \right\}$. Moreover, if $\Delta_1 \preccurlyeq \Delta_2$ with a full simulation relation, then (8) is equivalent to

$$\pi'(\mathcal{S}' \cap \mathcal{Z}_2) \subseteq \mathcal{S}', \tag{9}$$

where $\pi' : (0, z_2, w) \mapsto (0, 0, w)$ and $\mathcal{Z}_2 = \{(z_1, w, z_2) \mid z_1 = 0\}$.

Proof. By Proposition 1, $\Delta_1 \preccurlyeq \Delta_2$ if and only if there exists a subspace $\mathcal{S} \subseteq \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ such that (4) and (5) are satisfied. Define

$$\mathcal{S}' = \left\{ (z_1, w, z_2) \,\middle|\, \begin{bmatrix} z_1 \\ w \\ z_2 \\ w \end{bmatrix} \in \begin{bmatrix} \bar{L}_1 & 0 \\ L_1 & 0 \\ 0 & \bar{L}_2 \\ 0 & L_2 \end{bmatrix} \mathcal{S} \right\}.$$

It is clear by the definition of Δ'_{12} in (6) that $\mathcal{S}$ satisfies (4) if and only if $\mathcal{S}'$ satisfies (7).

Notice that under assumption (3), we have $E_1^2 = 0$, $E_2^2 = 0$, and E_1^1 and E_2^1 are of full column ranks. Indeed, let $\bar{M}_1$ and M_1 be two matrices such that $\begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix}^{-1} = \left[\begin{smallmatrix} \bar{M}_1 & M_1 \end{smallmatrix}\right]$. By $\operatorname{rank} \left[\begin{smallmatrix} E_1 \\ L_1 \end{smallmatrix}\right] = \operatorname{rank} \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix} = n_1$, we have $\ker E_1 = \ker \bar{L}_1$ and $\ker L_1 \cap \ker \bar{L}_1 = 0$. Moreover, by construction, we have $\operatorname{im} \bar{M}_1 = \ker L_1$ and $\operatorname{im} M_1 = \ker \bar{L}_1 = \ker E_1$. It follows that $E_1^1 = E\bar{M}_1$ is of full column rank and $E_1^2 = EM_1 = 0$. In a similar way, we can prove that E_2^1 is of full column rank and $E_2^2 = 0$.

Now condition (5) is equivalent to that for any $\left[\begin{smallmatrix} 0 \\ e_1 \end{smallmatrix}\right] \in \ker \left[\begin{smallmatrix} E_1^1 & 0 \end{smallmatrix}\right] \cap \begin{bmatrix} \bar{L}_1 \\ L_1 \end{bmatrix} \mathcal{V}_1^*$, there exist $\begin{bmatrix} z_1 \\ w \\ z_2 \\ w \end{bmatrix} \in \begin{bmatrix} \bar{L}_1 & 0 \\ L_1 & 0 \\ 0 & \bar{L}_2 \\ 0 & L_2 \end{bmatrix} \mathcal{S}$ and $\left[\begin{smallmatrix} 0 \\ e_2 \end{smallmatrix}\right] \in \ker \left[\begin{smallmatrix} E_2^1 & 0 \end{smallmatrix}\right] \cap \begin{bmatrix} \bar{L}_2 \\ L_2 \end{bmatrix} \mathcal{V}_2^*$ such that

$$\begin{bmatrix} 0 \\ e_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} z_1 \\ w \\ z_2 \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ e_2 \end{bmatrix}.$$

It follows that $z_1 = 0$, $z_2 = 0$ and $e_1 = w = -e_2$. Thus by the definitions of $\mathcal{S}'$, $\mathcal{V}'_1$ and $\mathcal{V}'_2$, condition (8) is equivalent to that for any $\begin{bmatrix} 0 \\ e_1 \\ 0 \end{bmatrix} \in \ker E_{12} \cap \mathcal{V}'_1$, there exist $\begin{bmatrix} 0 \\ w \\ 0 \end{bmatrix} \in \mathcal{S}'$ and $\begin{bmatrix} 0 \\ -e_2 \\ 0 \end{bmatrix} \in \ker E_{12} \cap \mathcal{V}'_2$ such that

$$\begin{bmatrix} 0 \\ e_1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ w \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ e_1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -e_2 \\ 0 \end{bmatrix}.$$

The latter condition is equivalent to

$$\ker E'_{12} \cap \mathcal{V}'_1 \subseteq \ker E'_{12} \cap \mathcal{V}'_2 \cap \mathcal{S}'.$$

Notice that because of (7) and the definition of $\mathcal{V}'_2$, we have $\mathcal{S}' \subseteq \mathcal{V}'_2$. It follows that

$$\mathcal{S}' \cap \mathcal{V}'_2 \cap \ker E'_{12} = \mathcal{S}' \cap \ker E'_{12}.$$

Then, it can be seen that (5) is equivalent to (8). Thus $\Delta_1 \preccurlyeq \Delta_2$ if and only if $\mathcal{S}'$ satisfies both (7) and (8). Moreover, if $\mathcal{S}$ is a full simulation relation, i.e., $\pi_{\mathcal{X}_1}(\mathcal{S}) = \mathcal{V}_1^*$, then

$$\mathcal{V}'_1 = \{(z_1, w, \tilde{z}_2) \mid (z_1, w, z_2) \in \mathcal{S}'\},$$

$$\ker E'_{12} \cap \mathcal{V}'_1 = \{(0, w, 0) \mid (0, w, z_2) \in \mathcal{S}'\}.$$

Hence the condition (8) is equivalent to (9).

□

Consider a DAE contract $\mathcal{C} = (\Delta_a, \Delta_g)$, assume that both Δ_a and Δ_g satisfy (3). Define

$$(E_{ag}, H_{ag}, L_{ag}) = \Delta_a \times \Delta_g$$

as the composition of the two DAEs. Define Π_g as a linear map which projects $(0, w, z_g, w) \mapsto (0, w, 0, w)$ and denote $T_{ag} = \begin{bmatrix} \bar{L}_a & 0 \\ L_a & 0 \\ 0 & \bar{L}_g \\ 0 & L_g \end{bmatrix}$. We have $L_{ag} = \begin{bmatrix} L_a & 0 \end{bmatrix}$ and denote $\bar{L}_{ag} = \begin{bmatrix} \bar{L}_a & 0 \end{bmatrix}$. Define two subspaces

$$\mathcal{V}^*_{ar} = \mathcal{V}^*_a \cap (L_a^{-1} L_{ag} \mathcal{R}) \text{ and } \bar{\mathcal{V}}^*_{ar} = \mathcal{V}^*_a \cap (L_a^{-1} L_{ag} \bar{\mathcal{R}}).$$

Theorem 1. *There exists an implementation Δ_m for the contract $\mathcal{C}$ if and only if there exist two subspaces $\bar{\mathcal{R}} \subseteq \mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_g$ such that*

(i) $\pi_{\mathcal{X}_a}(\mathcal{R}) = \mathcal{V}^*_{ar}$ *and* $\pi_{\mathcal{X}_a}(\bar{\mathcal{R}}) = \bar{\mathcal{V}}^*_{ar}$.
(ii) $H_{ag}\mathcal{R} \subseteq E_{ag}\mathcal{R}$ *and* $H_{ag}(\bar{\mathcal{R}} \cap \ker L_{ag}) \subseteq E_{ag}\bar{\mathcal{R}}$.
(iii) $P_g(\bar{\mathcal{R}} \cap \ker \bar{L}_{ag}) \subseteq \bar{\mathcal{R}}$, *where* $P_g = \Pi_g \cdot T_{ag}$.

Proof. Necessity: Suppose that the implementation $\Delta_m = (E_m, H_m, L_m)$ exists and $\Delta_a \times \Delta_m \preccurlyeq \Delta_g$ is a full simulation. Then there exists a simulation relation $\mathcal{S} \subseteq \mathcal{X}_a \times \mathcal{X}_m \times \mathcal{X}_g$ such that $\pi_{\mathcal{X}_a \times \mathcal{X}_m}(\mathcal{S}) = \mathcal{V}^*_{am}$, and

$$\begin{bmatrix} H_a & 0 & 0 \\ 0 & H_m & 0 \\ L_a & -L_m & 0 \\ 0 & 0 & H_g \\ L_a & 0 & -L_g \end{bmatrix} \mathcal{S} \subseteq \begin{bmatrix} E_a & 0 & 0 \\ 0 & E_m & 0 \\ 0 & 0 & 0 \\ 0 & 0 & E_g \\ 0 & 0 & 0 \end{bmatrix} \mathcal{S}$$

$$\begin{bmatrix} \ker \begin{bmatrix} E_a & 0 \\ 0 & E_m \end{bmatrix} \cap \mathcal{V}^*_{am} \\ 0 \end{bmatrix} \subseteq \mathcal{S} + \begin{bmatrix} 0 \\ \ker E_g \cap \mathcal{V}^*_g \end{bmatrix}.$$

Observe that for all $(x_a, x_m, x_g) \in \mathcal{S}$, we have $L_a x_a = L_m x_m = L_g x_g$. Find $\bar{L}_a$, $\bar{L}_m$, $\bar{L}_g$ such that we can define new coordinates

$$\begin{bmatrix} z_a \\ w \end{bmatrix} = \begin{bmatrix} \bar{L}_a \\ L_a \end{bmatrix} x_a, \ \begin{bmatrix} z_m \\ w \end{bmatrix} = \begin{bmatrix} \bar{L}_m \\ L_m \end{bmatrix} x_m, \ \begin{bmatrix} z_g \\ w \end{bmatrix} = \begin{bmatrix} \bar{L}_g \\ L_g \end{bmatrix} x_g. \tag{10}$$

Via changing of coordinates and reducing the shared variables, by Lemma 1, we have that the subspace $\mathcal{S}'$, which is $\mathcal{S}$ in the reduced coordinates, satisfies

$$\begin{bmatrix} H^1_a & 0 & 0 & H^2_a \\ 0 & H^1_m & 0 & H^2_m \\ 0 & 0 & H^1_g & H^2_g \end{bmatrix} \mathcal{S}' \subseteq \begin{bmatrix} E^1_a & 0 & 0 & E^2_a \\ 0 & E^1_m & 0 & E^2_m \\ 0 & 0 & E^1_g & E^2_g \end{bmatrix} \mathcal{S}' \tag{11}$$

$$\pi'(\mathcal{S}' \cap \mathcal{Z}'_g) \subseteq \mathcal{S}', \tag{12}$$

where

$$\mathcal{Z}'_g = \{(z_a, z_m, z_g, w) \mid z_a = 0, z_m = 0\}$$
$$\pi' : (0, 0, z_g, w) \mapsto (0, 0, 0, w).$$

Denote

$$\mathcal{V}'_{am} = \left\{ (z_a, z_m, w) \;\middle|\; \begin{bmatrix} z_a \\ w \\ z_m \\ w \end{bmatrix} \in \begin{bmatrix} \bar{L}_a & 0 \\ L_a & 0 \\ 0 & \bar{L}_m \\ 0 & L_m \end{bmatrix} \mathcal{V}^*_{am} \right\}$$

Define two subspaces

$$\mathcal{R}' := \{(z_a, z_g, w) \mid (z_a, z_m, z_g, w) \in \mathcal{S}'\} \tag{13}$$

and

$$\bar{\mathcal{R}}' := \{(\bar{z}_a, \bar{z}_g, \bar{w}) \mid (\bar{z}_a, 0, \bar{z}_g, \bar{w}) \in \mathcal{S}'\}. \tag{14}$$

It is clear that $\bar{\mathcal{R}}' \subseteq \mathcal{R}'$. Recall $\mathcal{S}$ is a full simulation relation, we have $\pi_{\mathcal{X}_a \times \mathcal{X}_m}(\mathcal{S}) = \mathcal{V}^*_{am}$. Thus

$$\pi_{amw}(\mathcal{S}') = \{(z_a, z_m, w) \mid (z_a, z_m, z_g, w) \in \mathcal{S}'\} = \mathcal{V}'_{am}.$$

Note that $\pi_w(\bar{\mathcal{R}}') \subseteq \pi_w(\mathcal{R}') = \pi_w(\mathcal{V}'_{am}) \subseteq \pi_w(\mathcal{V}'_a)$. Then from the above relations and by the definitions of $\mathcal{R}'$, $\bar{\mathcal{R}}'$ and $\mathcal{V}'_{am}$, we have

$$\begin{aligned} \pi_a(\mathcal{R}') &= \{z_a \mid (z_a, z_m, z_g, w) \in \mathcal{S}'\} \\ &= \{z_a \mid (z_a, z_m, w) \in \mathcal{V}'_{am},\ w \in \pi_w(\mathcal{R}')\} \\ &= \{z_a \mid (z_a, w) \in \mathcal{V}'_a,\ w \in \pi_w(\mathcal{R}')\}, \\ \pi_a(\bar{\mathcal{R}}') &= \{\bar{z}_a \mid (\bar{z}_a, \bar{w}) \in \mathcal{V}'_a,\ \bar{w} \in \pi_w(\bar{\mathcal{R}}')\}, \end{aligned} \tag{15}$$

where $\mathcal{V}'_a = \left\{ (z_a, w) \;\middle|\; \begin{bmatrix} z_a \\ w \end{bmatrix} \in \begin{bmatrix} \bar{L}_a \\ L_a \end{bmatrix} \mathcal{V}^*_a \right\}$.

Thus in the original coordinates, item (i) holds. Indeed, from the definition of subspace $\pi_a(\mathcal{R}')$ above, by $(z_a, w) \in \mathcal{V}'_a$ and $w \in \pi_w(\mathcal{R}')$, we have $x_a \in \mathcal{V}^*_a$ and $L_a x_a = w \in L_{ag}\mathcal{R}$. Therefore, we have $x_a \in \mathcal{V}^*_a \cap L_a^{-1} L_{ag}\mathcal{R}$, which implies $\pi_{\mathcal{X}_a}(\mathcal{R}) = \mathcal{V}^*_a \cap (L_a^{-1} L_{ag}\mathcal{R})$. With a similar proof, we can show that $\pi_{\mathcal{X}_a}(\bar{\mathcal{R}}) = \mathcal{V}^*_a \cap (L_a^{-1} L_{ag}\bar{\mathcal{R}})$.

Moreover, it can be seen from (11) that

$$H'_{ag}\mathcal{R}' \subseteq E'_{ag}\mathcal{R}',$$
$$H'_{ag}(\bar{\mathcal{R}}' \cap \ker L'_{ag}) \subseteq E'_{ag}\bar{\mathcal{R}}',$$

where $\ker L'_{ag} = \{(z_a, z_g, w) \mid w = 0\}$. So item (ii) is also satisfied. Finally, by (12), we have

$$\pi''(\bar{\mathcal{R}}' \cap \mathcal{Z}'_g) \subseteq \bar{\mathcal{R}}', \tag{16}$$

where $\pi'' : (0, z_g, w) \mapsto (0, 0, w)$ and $\mathcal{Z}'_g = \{(z_a, z_g, w) \mid z_a = 0\}$. Therefore, item (iii) holds.

Sufficiency: Now given two subspaces $\mathcal{R}$ and $\bar{\mathcal{R}}$ satisfying items (i)-(iii), we will construct the simulation relation $\mathcal{S}$ and the implementation Δ_m.

Step 1: Firstly, we can find the subspaces $\mathcal{R}'$ and $\bar{\mathcal{R}}'$ in the reduced coordinates from $\mathcal{R}$ and $\bar{\mathcal{R}}$ and (10). Let R' be a full column rank matrix such that $\operatorname{im} R' = \mathcal{R}'$ and denote

$$R' = \begin{bmatrix} R'_m \\ R'_w \end{bmatrix}.$$

Let $\bar{R}'$ be a full column rank matrix such that $\operatorname{im} \bar{R}' = \bar{\mathcal{R}}'$ and denote

$$\bar{R}' = \begin{bmatrix} \bar{R}'_m \\ \bar{R}'_w \end{bmatrix}.$$

Then define $z_{ag} = (z_a, z_g)$, $\bar{z}_{ag} = (\bar{z}_a, \bar{z}_g)$ and

$$\mathcal{S}' = \left\{ \begin{bmatrix} z_{ag} + \bar{z}_{ag} \\ w + \bar{w} \\ z_m \end{bmatrix} \;\middle|\; \begin{array}{l} \begin{bmatrix} z_{ag} \\ w \end{bmatrix} \in \mathcal{R}', \quad \begin{bmatrix} \bar{z}_{ag} \\ \bar{w} \end{bmatrix} \in \bar{\mathcal{R}}', \\ R'_m z_m = z_{ag} \end{array} \right\}.$$

Because $H'_{ag}(\bar{\mathcal{R}}' \cap \ker L'_{ag}) \subseteq E'_{ag}\bar{\mathcal{R}}'$ and $\bar{\mathcal{R}}' \subseteq \mathcal{R}'$, via similar constructions for conditioned invariant subspaces in linear geometric control theory (see e.g., Theorem 5.5 of [19], also Theorem 3 of [8]), we can find matrices K_1 and K_2 such that

$$\begin{aligned} (H'_{ag} + K_1 L'_{ag})\bar{\mathcal{R}}' &\subseteq E'_{ag}\bar{\mathcal{R}}' \\ (H'_{ag} + K_2 L'_{ag})\bar{\mathcal{R}}' &\subseteq E'_{ag}\mathcal{R}', \end{aligned} \tag{17}$$

where

$$\left[\begin{array}{c|c} H'_{ag} & E'_{ag} \\ L'_{ag} & 0 \end{array}\right] = \left[\begin{array}{cc|cc} H^1_{ag} & H^2_{ag} & E^1_{ag} & 0 \\ 0 & I & 0 & 0 \end{array}\right].$$

Notice that by (17), we have

$$(K_2 - K_1)L'_{ag}\bar{\mathcal{R}}' = (K_2 - K_1)\operatorname{im} R'_w \subseteq E'_{ag}\mathcal{R}'.$$

In order to satisfy the inclusion (11), we will construct Δ_m via (where z_m and z_g are permuted)

$$\begin{bmatrix} H^1_{ag} & H^2_{ag} & 0 \\ 0 & H^2_m & H^1_m \end{bmatrix} \mathcal{S}' \subseteq \begin{bmatrix} E^1_{ag} & 0 & 0 \\ 0 & 0 & E^1_m \end{bmatrix} \mathcal{S}'.$$

Let

$$\begin{aligned} \tilde{E}^1_m &= E^1_{ag} R'_m, \quad \tilde{H}^2_m = K_2 - K_1, \\ \tilde{H}^1_m &= H^1_{ag} R'_m + (H^2_{ag} - K_2 + K_1) R'_w. \end{aligned} \tag{18}$$

Then find a full row rank matrix $\bar{H}^2_m$ with $\operatorname{rank} \bar{H}^2_m = \dim w - \operatorname{rank} \bar{R}'_w$ and a matrix $\bar{H}^1_m$ such that

$$\begin{aligned} 0 &= \bar{H}^2_m \bar{R}'_w, \\ 0 &= \bar{H}^1_m + \bar{H}^2_m R'_w. \end{aligned} \tag{19}$$

We define

$$\begin{bmatrix} E^1_m & H^1_m & H^2_m \end{bmatrix} := \begin{bmatrix} \tilde{E}^1_m & \tilde{H}^1_m & \tilde{H}^2_m \\ 0 & \bar{H}^1_m & \bar{H}^2_m \end{bmatrix}. \tag{20}$$

Step 2: Now we show that $\Delta_m = (E_m, H_m, L_m)$ is an implementation of the contract $\mathcal{C}$, where

$$E_m = \begin{bmatrix} E^1_m & 0 \end{bmatrix}, \quad H_m = \begin{bmatrix} H^1_m & H^2_m \end{bmatrix}, \quad L_m = \begin{bmatrix} 0 & I \end{bmatrix}. \tag{21}$$

Indeed, as (11) is satisfied by construction, we only need to show $\pi_{am}(\mathcal{S})' = \mathcal{V}'_{am}$ and (12) holds. By the construction of Δ_m, we have

$$\mathcal{V}'_{am} = \left\{ \begin{bmatrix} z_a + \bar{z}_a \\ z_m \\ w + \bar{w} \end{bmatrix} \;\middle|\; \begin{bmatrix} z_a + \bar{z}_a \\ w + \bar{w} \end{bmatrix} \in \mathcal{V}'_a,\ z_m = \pi_m(\mathcal{S}') \right\}$$
$$w \in \pi_w(\mathcal{R}'), \bar{w} \in \pi_w(\bar{\mathcal{R}}')$$

Then by item (i), or equivalently, (15), we have

$$\begin{aligned} \mathcal{V}'_{am} &= \left\{ \begin{bmatrix} z_a + \bar{z}_a \\ z_m \\ w + \bar{w} \end{bmatrix} \;\middle|\; \begin{bmatrix} z_a \\ w \end{bmatrix} \in \pi_{aw}(\mathcal{R}'),\ \begin{bmatrix} \bar{z}_a \\ \bar{w} \end{bmatrix} \in \pi_{aw}(\bar{\mathcal{R}}') \right\} \\ &\quad z_m = \pi_m(\mathcal{S}') \\ &= \pi_{amw}(\mathcal{S}'). \end{aligned}$$

Step 3: As $\bar{\mathcal{R}}$ satisfies item (iii), we have that (16) holds, i.e.,

$$\left\{ (0, 0, w) \;\middle|\; (0, z_g, w) \in \bar{\mathcal{R}}' \right\} \subseteq \bar{\mathcal{R}}'.$$

Because $\bar{\mathcal{R}}'$ satisfies (14) by the construction of $\mathcal{S}'$, we get

$$\{ (0, 0, 0, w) \mid (0, 0, z_g, w) \in \mathcal{S}' \} \subseteq \mathcal{S}',$$

which means (12) holds. Hence the subspace $\mathcal{S}$, which is $\mathcal{S}'$ in the original (x_a, x_m, x_g)-coordinates, is a full simulation relation of $\Delta_a \times \Delta_m$ by Δ_g. □

Remark 3. Condition (i) of Theorem 1 follows from the fact that $\mathcal{S}$ is a full simulation relation. Condition (ii) follows from condition (4), or equivalently, condition (7). Similarly, condition (iii) follows from condition (5) or (9).

Remark 4. Theorem 1 is not only a DAE generalization of Theorem 5 in [14] for ODE control systems in the absence of disturbances, but also exhibits several substantial differences. First, all DAE components are represented in a unified form, with interconnections established by sharing variables. This formulation offers greater flexibility, without restricting the components to specific input-output forms or interconnection structures. For instance, the environment and implementation in [14] are interconnected in an input-output series configuration. Consequently, the implementation does not affect the dynamics of the environment, leading to a simpler condition (i), namely, $\pi_{\mathcal{X}_a}(\mathcal{R}) = \pi_{\mathcal{X}_a}(\bar{\mathcal{R}}) = \mathcal{V}^*_a$. This is not the case in our setting. Second, controlled and conditioned invariants are replaced by DAE invariants, an approach supported by the results in [5]. Third, both $\mathcal{R}$ and $\bar{\mathcal{R}}$ can be constructed more systematically using Wong sequences or their generalizations; see also the discussion below and the comparison with [3].

B. Algorithm for constructing DAE contract implementation

As the consistency space $\mathcal{V}^*_m$ of any implementation Δ_m is related to the two subspace $\mathcal{R}$ and $\bar{\mathcal{R}}$, the first step of finding the implementation is to construct the largest $\mathcal{R}$ and $\bar{\mathcal{R}}$. Clearly, the consistency space

$$\mathcal{R} = \mathcal{V}^*_{ag}$$

of $\Delta_a \times \Delta_g$ is the largest subspace such that $H_{ag}\mathcal{R} \subseteq E_{ag}\mathcal{R}$. Then we propose the following fix-point algorithm to calculate the largest subspace $\bar{\mathcal{R}} \subseteq \mathcal{R}$ such that conditions (ii) and (iii) of Theorem 1 hold, define

$$\begin{aligned} &\bar{\mathcal{R}}^0 = \mathcal{R}, \\ &\bar{\mathcal{R}}^{j+1} = \bar{\mathcal{R}}^j \cap (H^{-1}_{ag} E_{ag} \bar{\mathcal{R}}^j + \operatorname{im} L^T_{ag}) \cap (P^{-1}_{ag} \bar{\mathcal{R}}^j + \operatorname{im} \bar{L}^T_{ag}) \end{aligned} \tag{22}$$

For the sequence $\bar{\mathcal{R}}^0, \ldots, \bar{\mathcal{R}}^j$, there exists a smallest number $0 \le j^* \le \dim \mathcal{R}$ such that $\bar{\mathcal{R}}^{j^*+1} = \bar{\mathcal{R}}^{j^*}$, then it is seen that

$$\bar{\mathcal{R}} = \bar{\mathcal{R}}^{j^*}$$

is the largest subspace in $\mathcal{R}$ and satisfies (ii) and (iii).

Then we propose Algorithm 1 below for constructing an implementation Δ_m based on $\mathcal{R}$ and $\bar{\mathcal{R}}$ given above.

The following proposition shows that the constructed implementation Δ_m is the largest among all the implementations with shared variables in $\pi_w(\mathcal{V}^*_a) = L_a \mathcal{V}^*_a$.

Proposition 4. *Consider a contract $\mathcal{C} = (\Delta_a, \Delta_g)$. Assume that the constructed subspaces $\mathcal{R}$ and $\bar{\mathcal{R}}$ satisfy conditions* (i) *of* Theorem 1. *Then* Algorithm 1 *outputs a DAE Δ_m satisfying*

(i) $\Delta_m \in \mathcal{M}_{\mathcal{C}}$.
(ii) $\forall \tilde{\Delta}_m \in \mathcal{M}_{\mathcal{C}}$ *such that* $\tilde{L}_m \tilde{\mathcal{V}}^*_m \subseteq L_a \mathcal{V}^*_a$, *we have* $\tilde{\Delta}_m \preccurlyeq \Delta_m$ *with a full simulation relation.*

Proof. Item (i) is proved in the sufficiency part of Theorem 1 above. Then consider any DAE $\tilde{\Delta}_m \in \mathcal{M}_{\mathcal{C}}$ satisfying $\tilde{L}_m \tilde{\mathcal{V}}^*_m = \pi_w(\tilde{\mathcal{V}}^*_m) \subseteq \pi_w(\mathcal{V}^*_a) = L_a \mathcal{V}^*_a$.

Step 1: We define

$$\hat{\mathcal{S}} = \left\{ (\tilde{z}_m, w, z_m) \;\middle|\; \begin{bmatrix} \tilde{z}_m \\ w \end{bmatrix} \in \mathcal{V}'(\tilde{\Delta}_m),\ \begin{bmatrix} z_m \\ w \end{bmatrix} \in \mathcal{V}'(\Delta_m) \right\},$$

where $\mathcal{V}'(\tilde{\Delta}_m)$ and $\mathcal{V}'(\Delta_m)$ denote, respectively, the consistency space of $\tilde{\Delta}_m$ in $(\tilde{z}_m, w)$-coordinates and that of Δ_m in (z_m, w)-coordinates. Clearly, $\hat{\mathcal{S}}$ is a linear subspace and it satisfies the inclusion (7) for $\Delta_1 = \tilde{\Delta}_m$ and $\Delta_2 = \Delta_m$.

Step 2: Denote the simulation relation of $\Delta_a \times \Delta_m \preccurlyeq \Delta_g$ by $\mathcal{S}'(\Delta_m)$, and that of $\Delta_a \times \tilde{\Delta}_m \preccurlyeq \Delta_g$ by $\mathcal{S}'(\tilde{\Delta}_m)$. Define

Algorithm 1 DAE contract implementation

Input: $\mathcal{C} = (\Delta_a, \Delta_g)$,
Output: Δ_m implements $\mathcal{C}$

Define (E_{ag}, H_{ag}, L_{ag})=$\Delta_a \times \Delta_g$.
Let $\mathcal{R} = \mathcal{V}^*_{ag}$. Then construct $\bar{\mathcal{R}} = \bar{\mathcal{R}}^{j^*}$ by (22).
Find $\mathcal{V}^*_{ar}$ and $\bar{\mathcal{V}}^*_{ar}$.
if $\pi_{\mathcal{X}_a}(\mathcal{R}) \neq \mathcal{V}^*_{ar}$ or $\pi_{\mathcal{X}_a}(\bar{\mathcal{R}}) \neq \bar{\mathcal{V}}^*_{ar}$. **then**
  Stop
end if
Find subspaces $\mathcal{R}'$ and $\bar{\mathcal{R}}'$, which are, respectively, $\mathcal{R}$ and $\bar{\mathcal{R}}$ in the (z_a, z_g, w)-coordinates defined by (10).
Let $R' = \begin{bmatrix} R'_m \\ R'_w \end{bmatrix}$ be a full column rank matrix such that $\operatorname{im} R' = \mathcal{R}'$ and $\bar{R}' = \bar{\mathcal{R}}'$ be a full column rank matrix such that $\operatorname{im} \bar{R}' = \bar{\mathcal{R}}'$.
Construct matrices E'_{ag}, H'_{ag} and L'_{ag}, which are E_{ag}, H_{ag} and L_{ag} the (z_a, z_g, w)-coordinates, respectively.
Find matrices K_1 and K_2 such that (17) holds.
Construct $\Delta_m = (E_m, H_m, L_m)$ by (18), (20), (21).
if $\tilde{E}^1_m$ is of full column rank **then**
  Replace $\tilde{E}^1_m$ by I, $\tilde{H}^1_m$ by $(\tilde{E}^1_m)^\dagger \tilde{H}^1_m$ and $\tilde{H}^2_m$ by $(\tilde{E}^1_m)^\dagger \tilde{H}^2_m$, where $(\tilde{E}^1_m)^\dagger$ is a left-inverse of $\tilde{E}^1_m$.
end if

the subspaces $\mathcal{R}'(\Delta_m)$, $\bar{\mathcal{R}}'(\Delta_m)$ and $\mathcal{R}'(\tilde{\Delta}_m)$, $\bar{\mathcal{R}}'(\tilde{\Delta}_m)$ from $\mathcal{S}'(\Delta_m)$ and $\mathcal{S}'(\tilde{\Delta}_m)$, respectively, via similar definitions as in (13) and (14). Because $\mathcal{R}'(\Delta_m) \cong \mathcal{R}$ and $\bar{\mathcal{R}}'(\Delta_m) \cong \bar{\mathcal{R}}$ are the largest subspaces constructed to satisfy the conditions of Theorem 1, we have

$$\mathcal{R}'(\tilde{\Delta}_m) \subseteq \mathcal{R}'(\Delta_m), \quad \bar{\mathcal{R}}'(\tilde{\Delta}_m) \subseteq \bar{\mathcal{R}}'(\Delta_m).$$

Step 3: Because $\mathcal{S}'(\Delta_m)$ and $\mathcal{S}'(\tilde{\Delta}_m)$ are both full simulation relations, with a similar argument as in the proof of Theorem 1, we have $\pi_w(\mathcal{R}'(\Delta_m)) = \pi_w(\mathcal{V}'(\Delta_{am}))$ and $\pi_w(\mathcal{R}'(\tilde{\Delta}_m)) = \pi_w(\mathcal{V}'(\tilde{\Delta}_{am}))$. Observe that the condition $\pi_w(\mathcal{V}^*_m) \subseteq \pi_w(\mathcal{V}^*_a)$ in item(ii) is equivalent to $\pi_w(\mathcal{V}'(\tilde{\Delta}_m)) \subseteq \pi_w(\mathcal{V}'(\Delta_a))$. It follows that

$$\begin{aligned}
\pi_w(\mathcal{V}'(\tilde{\Delta}_{am})) &= \left\{ w \;\middle|\; (z_a, \tilde{z}_m, w) \in \mathcal{V}'(\tilde{\Delta}_{am}) \right\} \\
&= \left\{ w \;\middle|\; \begin{bmatrix} z_a \\ w \end{bmatrix} \in \mathcal{V}'(\Delta_a),\ \begin{bmatrix} \tilde{z}_m \\ w \end{bmatrix} \in \mathcal{V}'(\tilde{\Delta}_m) \right\}. \\
&= \left\{ w \;\middle|\; \begin{bmatrix} \tilde{z}_m \\ w \end{bmatrix} \in \mathcal{V}'(\tilde{\Delta}_m) \right\} = \pi_w(\mathcal{V}'(\tilde{\Delta}_m))
\end{aligned}$$

Recall the construction of Δ_m, we get $\pi_w(\mathcal{R}'(\Delta_m)) = \pi_w(\mathcal{V}'(\Delta_m))$. Consequently, we have

$$\begin{aligned}
\pi_w(\mathcal{V}'(\tilde{\Delta}_m)) &= \pi_w(\mathcal{V}'(\tilde{\Delta}_{am})) = \pi_w(\mathcal{R}'(\tilde{\Delta}_m)) \\
&\subseteq \pi_w(\mathcal{R}'(\Delta_m)) = \pi_w(\mathcal{V}'(\Delta_m)).
\end{aligned}$$

The above relation implies

$$\pi_{\tilde{z}_m, w}(\hat{\mathcal{S}}) = \mathcal{V}'(\tilde{\Delta}_m),$$

which is the requirement that $\hat{S}$ is a full simulation relation.

Step 4: Take any vector $(0, w) \in \mathcal{V}'(\tilde{\Delta}_m)$, it can be seen that $w \in \pi_w(\bar{\mathcal{R}}'(\tilde{\Delta}_m))$ by definition. Thus by $\pi_w(\bar{\mathcal{R}}'(\tilde{\Delta}_m)) \subseteq \pi_w(\bar{\mathcal{R}}'(\Delta_m))$, we have $w \in \pi_w(\bar{\mathcal{R}}'(\Delta_m))$. It follows that $(0, w) \in \mathcal{V}'(\Delta_m)$. This condition implies that (8) holds for $\mathcal{S}' = \hat{\mathcal{S}}$, $\Delta_1 = \tilde{\Delta}_m$ and $\Delta_2 = \Delta_m$. Hence by Lemma 1, we have $\tilde{\Delta}_m \preccurlyeq \Delta_m$ with a full simulation relation.

□

V. Example

Consider two electrical circuits A and G as in Figures 1 and 2, respectively. The model of Circuit A is $\Delta_a = (E_a, H_a)$:

$$\begin{bmatrix} C_a & 0 & 0 & 0 & 0 \\ 0 & L_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}^1_a \\ \dot{z}^2_a \\ \dot{w}_0 \\ \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} \frac{-1}{R_a} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} z^1_a \\ z^2_a \\ w_0 \\ w_1 \\ w_2 \end{bmatrix},$$

the model of Circuit G is $\Delta_g = (E_g, H_g)$:

$$\begin{bmatrix} C^1_g & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L^2_g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C^3_g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L^4_g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}^1_g \\ \dot{z}^2_g \\ \dot{z}^3_g \\ \dot{z}^4_g \\ \dot{w}_0 \\ \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} \frac{-1}{R^1_g} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -\frac{1}{R^4_g} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} z^1_g \\ z^2_g \\ z^3_g \\ z^4_g \\ w_0 \\ w_1 \\ w_2 \end{bmatrix},$$

the shared variables are $w = (w_0, w_1, w_2)$. For the ease of calculation, we set

$$\begin{aligned} C_a = C^1_g = C^3_g = 1F, \quad L_a = L^2_g = L^4_g = 1H, \\ R_a = R^1_g = 1\Omega, \quad R^4_g = 0.5\Omega. \end{aligned}$$

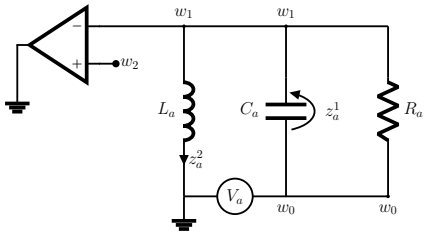


Fig. 1: Circuit A

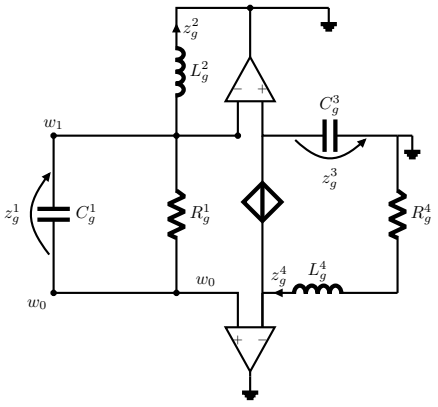


Fig. 2: Circuit G

Define a DAE contract $\mathcal{C} = (\Delta_a, \Delta_g)$, the aim is to find an implementation of $\mathcal{C}$ via Algorithm 1. First we construct

the composition $\Delta_a \times \Delta_g$ and find $\Delta'_{ag} \overset{\bullet}{=} \Delta_a \times \Delta_g$ in (z_a, z_g, w)-coordinates. By (2) and (22), we find

$$\mathcal{R}' = \mathcal{V}'_{ag} = \mathrm{im} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathcal{R}}' = \mathrm{im} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

It can be checked that condition (i) of Theorem 1 is satisfied in (z_a, z_g, w)-coordinates. Moreover, via (18), (20) and (21), we find

$$\Delta_m : \begin{cases} \dot{z}^1_m = z^2_m - w_0 \\ \dot{z}^2_m = z^3_m \\ \dot{z}^4_m = z^3_m \\ \dot{z}^3_m = -2z^3_m - w_0 \\ 0 = z^3_m - w_1 \\ 0 = z^4_m - w_2 \end{cases} \overset{\bullet}{=} \begin{cases} \dot{z}^4_m = z^3_m \\ \dot{z}^3_m = -2z^3_m - w_0 \\ 0 = z^4_m - w_1 \\ 0 = w_1 - w_2 \end{cases}.$$

A circuit realization of the DAE Δ_m is Circuit M as shown in Figure 3 with $C^4_m = 1F$, $L^3_m = 1H$ and $R^3_m = 0.5\Omega$.

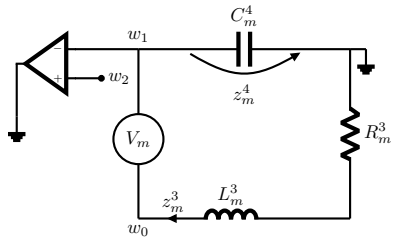


Fig. 3: Circuit M

VI. Conclusions and discussions

In this paper, we investigate DAE-based assumption-guarantee contracts with components in a unified linear DAE form. First, we define the composition of two DAEs and analyze its properties in terms of simulation relations. We then formalize the notions of implementation and environment for DAE contracts and establish necessary and sufficient conditions for the existence of implementations. Furthermore, we present an algorithm for constructing an implementation. Finally, we demonstrate the effectiveness of the proposed algorithm through an illustrative example involving electrical circuits.

References

[1] 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," *Found. Trends Electron. Des. Autom.*, vol. 12, no. 2-3, pp. 124–400, 03 2018.

[2] T. Berger, A. Ilchmann, and S. Trenn, "The quasi-Weierstraß form for regular matrix pencils," *Linear Algebra Appl.*, vol. 436, no. 10, pp. 4052–4069, 2012.

[3] T. Berger, T. Reis, and S. Trenn, "Observability of linear differential-algebraic systems: A survey," in *Surveys in Differential-Algebraic Equations IV*. Springer, 2017, pp. 161–219.

[4] B. Besselink, K. H. Johansson, and A. Van Der Schaft, "Contracts as specifications for dynamical systems in driving variable form," in *European Control Conf. (ECC)*, 2019, pp. 263–268.

[5] Y. Chen and W. Respondek, "Geometric analysis of linear differential-algebraic equations via linear control theory," *SIAM J. Control Optim.*, vol. 59, no. 1, pp. 103–130, 2021.

[6] L. Dai, *Singular Control Systems*, ser. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer, 1989, vol. 118.

[7] F. L. Lewis, "A survey of linear singular systems," *Circuits, Systems and Signal Processing*, vol. 5, no. 1, pp. 3–36, 1986.

[8] ——, "Geometric design techniques for observers in singular systems," *Automatica*, vol. 26, no. 2, pp. 411–415, 1990.

[9] N. Y. Megawati and A. v. d. Schaft, "Bisimulation equivalence of differential-algebraic systems," *Int. J. Control*, vol. 91, no. 1, pp. 45–56, 2018.

[10] I. X. I. Romeo, "The Algebra of Contracts," Ph.D. dissertation, University of California, Berkeley, 2022.

[11] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems," *European J. Control*, vol. 18, no. 3, pp. 217–238, 2012.

[12] A. Saoud, A. Girard, and L. Fribourg, "Assume-guarantee contracts for continuous-time systems," *Automatica*, vol. 134, p. 109910, 2021.

[13] B. M. Shali, A. van der Schaft, and B. Besselink, "Composition of behavioural assume–guarantee contracts," *IEEE Trans. Autom. Control*, vol. 68, no. 10, pp. 5991–6006, 2022.

[14] B. M. Shali, A. J. van der Schaft, and B. Besselink, "Design and control for implementation of simulation-based assume-guarantee contracts," *IEEE Trans. Autom. Control*, vol. 71, no. 1, pp. 65–80, 2025.

[15] ——, "Design and control for implementation of simulation-based assume-guarantee contracts," *IEEE Trans. Autom. Control*, vol. 71, no. 1, pp. 65–80, 2026.

[16] M. Sharf, B. Besselink, and K. H. Johansson, "Contract composition for dynamical control systems: Definition and verification using linear programming," *Automatica*, vol. 164, p. 111637, 2024.

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

[18] S. Trenn, "Distributional differential algebraic equations," Ph.D. dissertation, Institut für Mathematik, Technische Universität Ilmenau, Ilmenau, Germany, 2009.

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

[20] A. J. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 2160–2172, 2004.

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

[22] K.-T. Wong, "The eigenvalue problem $\lambda Tx + Sx$," *J. Diff. Eqns.*, vol. 16, pp. 270–280, 1974.