Formal Control Synthesis via Simulation Relations and Behavioural Theory for Discrete-time Descriptor Systems

Sofie Haesaert, *Member, IEEE*, Fei Chen, Alessandro Abate, *Senior Member, IEEE*, and Siep Weiland

***Abstract*—The control and verification of industrial processes, modelled as discrete-time descriptor systems, is often computationally hard due to the presence of both algebraic couplings and difference equations. In this paper, we introduce a new control synthesis method for descriptor systems that is based on formal abstractions and enables control design over related reduced-order models. We leverage notions of exact and approximate similarity relations, which hold for the algebraic couplings that are inherent to descriptor systems. Using the behavioural framework, we extend a control refinement scheme for classical dynamical systems and develop a corresponding notion for descriptor systems: we show that any given well-posed controller of the abstract (reduced-order) descriptor system can be refined to a controller for the original descriptor system. The resulting controlled system preserves the same controlled output behaviour in the case of exact similarity, whereas in the case of approximate similarity the output behaviour of the controlled descriptor system is shown to have a bounded deviation from that of the abstract model where the controller is designed.**

***Index Terms*—Descriptor systems, behavioural theory, approximate simulation relations, formal verification.**

I. Introduction

COMPLEX industrial processes can generally encompass algebraic couplings in addition to differential (or difference) equations of high order. Models for these processes, which are referred to as descriptor systems [9], [19], are commonly used to describe mechanical systems. These algebraic equations and couplings, together with large state space dimensions, render numerical simulation and controller design challenging. In particular, the presence of algebraic equations hinders the verification and synthesis of controllers with respect to formal specifications, such as linear temporal logic properties [4], [20], [23]. For systems solely comprised of difference or differential equations, the use of formal abstractions enables their automated verification [23] and formal controller synthesis, namely the automated proof of pre-specified requirements or specifications, and the algorithmic design of certifiable correct-by-design controllers [18], [23]. With focus on control design tasks, methods based on formal abstractions first reduce the original (concrete) system to an *abstract system* with a finite or smaller-dimensional state space, over which a controller can be synthesised. The controller obtained for the abstract system is then *refined* over the concrete system by leveraging the existence of an (approximate) *simulation relation* between the two systems [14], [23]. This step is referred to as *control refinement*.

Model order reduction techniques can be used together with formal abstractions in a control refinement framework, as presented in [13] for models described by ordinary differential or difference equations. Model reduction methods [1] are used to replace concrete systems with simpler, reduced-order models: even though most reduction methods have been developed for standard dynamical models, recent research has also targeted descriptor systems [6]. However, the connection between model order reduction techniques and formal abstractions presents a fundamental challenge for descriptor systems: the presence of anti-causality in the algebraic equations of discrete-time descriptor systems [9] makes the standard control refinement via simulation relations impossible.

In this paper, we specifically consider the control refinement problem for discrete-time descriptor systems via simulation relations within a behavioural framework, such that properties verified over the future behaviour of the abstract system are also verified over the concrete controlled system. Within the behavioural theory [25], [27], [29], [30], a formal distinction is made between a system (its behaviour) and its representations, enabling us to investigate descriptor systems and control refinement problems without having to deal with their inherent anti-causality (mentioned earlier) directly. Additionally within this framework, control designs are not limited to causal memory-dependent mappings from observed state to input, but can include more general nonlinear and algebraic equations. We leverage (approximate) simulation relations [23] to construct refined controllers that satisfy both general well-posedness notions and regularity conditions on their control interconnection.

Literature review: The development of simulation relations between descriptor systems has been recently investigated in [22]: this work deals with continuous-time descriptor systems and focuses on conditions for bisimilarity and on the construction of exact similarity relations. Limited to verification applications, [22] does not consider control refinement. Additionally, the relations used in [22] are subspaces, therefore it cannot easily be extended to the approximate case nor to nonlinear algebraic differential equations. The results presented in this paper for discrete-time systems carry over to continuous-time systems and are therefore complementary to

Sofie Haesaert (s.haesaert@tue.nl) and Siep Weiland (s.weiland@tue.nl) are with the Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands.

Fei Chen (fchen@kth.se) is with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden.

Alessandro Abate (aabate@cs.ox.ac.uk) is with the Department of Computer Science, University of Oxford, United Kingdom.

results in [22].

In [7], preliminary results on the exact simulation relations have been presented. This paper extends those results with the unpublished proofs, and it fully develops the results for the approximate notion with specific examples. Additionally, it refines the analysis and construction of controllers to those that are regular [28].

Structure: The structure of this paper is as follows. We close this section with some mathematical notations used throughout this work. In the next section, we define the notion of discrete-time dynamical systems within a behavioural framework and use it to formalise the control refinement problem. Subsequently, Section III and Section IV discuss, respectively, exact/approximate simulation relations and a system transformation. Section V uses these notions to develop the exact control refinement scheme. In a similar way, the approximate control refinement is developed in Section VI. Section VII provides more details on the automated computations of controllers, and develops an illustrative case study.

Notations: The Euclidean norm is denoted by $\|\cdot\|$ and satisfies the triangle inequality $\|x+y\| \leq \|x\| + \|y\|$. The metric over the Euclidean space induced by $\|\cdot\|$ is denoted by $d(x_1, x_2) = \|x_1 - x_2\|$. A discrete-time signal u, that is $u : \mathbb{T} \to \mathbb{R}^m$ with $\mathbb{T} := \{0, 1, 2, 3, \ldots\}$, has a supremum norm denoted by $u_{\max}$ and defined as $u_{\max} = \sup_{t\in\mathbb{T}} \|u(t)\|$. Given an Euclidean space $\mathbb{X}$, the ε-ball $B_\varepsilon(x)$ of radius $\varepsilon > 0$ with centre $x \in \mathbb{X}$ is defined as $B_\varepsilon(x) = \{y \in \mathbb{X} \mid \|x - y\| \leq \varepsilon\}$. For a set $A \subset \mathbb{X}$, $\mathcal{C}_\varepsilon(A) = \{x \in \mathbb{X} \mid B_\varepsilon(x) \subseteq A\}$ is called the ε-contraction of A and $\mathcal{E}_\varepsilon(A) = \{x \in \mathbb{X} \mid B_\varepsilon(x) \cap A \neq \emptyset\}$ is called the ε-expansion of A. We say that a discrete-time signal $y : \mathbb{T} \to \mathbb{Y}$ is an element of $\mathbb{Y}^\mathbb{T}$. Over this product space $\mathbb{Y}^\mathbb{T}$, we trivially extend the ε-expansion as $\mathcal{E}_\varepsilon(y) = \{\tilde{y} \in \mathbb{Y}^\mathbb{T} \mid \forall t \in \mathbb{T} : \tilde{y}(t) \in \mathcal{E}_\varepsilon(y(t))\}$.

For two sets $\mathbb{X}_1$ and $\mathbb{X}_2$, the Cartesian product is defined as $\mathbb{X}_1 \times \mathbb{X}_2 = \{(x_1, x_2) \mid x_1 \in \mathbb{X}_1, x_2 \in \mathbb{X}_2\}$. A binary relation $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ is a subset of this Cartesian product that defines a relation among the elements $x_1 \in \mathbb{X}_1$ and $x_2 \in \mathbb{X}_2$.

II. Systems in a Behavioural Framework

As introduced in [30], we define discrete-time dynamical systems as follows. *A dynamical system $\mathbf{\Sigma}$ is defined as a triple*

$$\mathbf{\Sigma} = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$$

with $\mathbb{T}$ a subset of $\mathbb{Z}$, called the time axis, $\mathbb{W}$ a set called the signal space, and $\mathfrak{B}$ a subset of $\mathbb{W}^\mathbb{T}$ called the behaviour. Here $\mathbb{W}^\mathbb{T}$ is the notation for the collection of all maps from $\mathbb{T}$ to $\mathbb{W}$. The behaviour $\mathfrak{B}$ of a system is a set of trajectories or time-dependent functions that collect the trajectories that are compatible with the system [30]. This definition encapsulates both formal languages and descriptor systems, as defined next. (See [26]).

A. Discrete-time descriptor systems (DS)

For discrete-time systems, the time axis is defined as $\mathbb{T} := \mathbb{N} = \{0, 1, 2, \ldots\}$ and initialised at $t = 0$. Consider a discrete-time linear descriptor system (DS) $\mathbf{\Sigma}$ whose dynamics are defined by the tuple (E, A, B, C) as

$$\begin{aligned} Ex(t+1) &= Ax(t) + Bu(t); \\ y(t) &= Cx(t), \quad x(0) \in \mathbb{X}_0, \end{aligned} \tag{1}$$

with state $x(t) \in \mathbb{X} \subseteq \mathbb{R}^n$, input $u(t) \in \mathbb{U} \subseteq \mathbb{R}^p$, output $y(t) \in \mathbb{Y} \subseteq \mathbb{R}^k$ and initial state $x(0)$ restricted to $\mathbb{X}_0 \subseteq \mathbb{X}$. Further, $E, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times p}$ and $C \in \mathbb{R}^{k\times n}$ are constant matrices and we presume that $\operatorname{rank}(B) = p$ and $\operatorname{rank}(C) = k$. If E is non-singular, we refer to the corresponding dynamical system as a *non-singular DS*. In that case, we can transform (1) into standard state space equations, with $\tilde{E} = I$, $\tilde{A} = E^{-1}A, \tilde{B} = E^{-1}B$, and $\tilde{C} = C$.

We say that a trajectory $w = (u, x, y)$, with $w : \mathbb{N} \to (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$, satisfies (1) if for all $t \in \mathbb{N}$ the equations in (1) evaluated at $u(t), x(t), x(t+1), y(t)$ hold. The collection of all trajectories w defines the *full* behaviour, or equivalently the *input-state-output* behaviour associated with (1) as

$$\mathfrak{B}_{\mathrm{i/s/o}} := \{(u, x, y) \in (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})^\mathbb{N} | (1) \text{ is satisfied}\}. \tag{2}$$

The *manifest* behaviour or the *input/output* behaviour is given by

$$\begin{aligned} \mathfrak{B}_{\mathrm{i/o}} := \{&(u, y) \in (\mathbb{U} \times \mathbb{Y})^\mathbb{N} \mid \exists x \in \mathbb{X}^\mathbb{N} \\ &\text{s.t. } (u, x, y) \in \mathfrak{B}_{\mathrm{i/s/o}}\}. \end{aligned} \tag{3}$$

We say that a trajectory $w : \mathbb{N} \to (\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$ is initialised with $\mathbb{X}_0$ if (1) holds and $x(0) = x_0 \in \mathbb{X}_0$. These trajectories are also called the *continuations* of x_0. The collection of continuations with $x_0 \in \mathbb{X}_0$ is referred to as the initialised behaviour $\mathfrak{B}^{\mathrm{init}}_{\mathrm{i/s/o}}$. Let us formally define a discrete-time linear descriptor system as follows.

Definition 1. A (discrete-time) descriptor system is defined as a dynamical system $\mathbf{\Sigma}$ initialised with $\mathbb{X}_0$, whose behaviour can be represented by the combination of algebraic equations and difference equations (1), i.e.

$$\mathbf{\Sigma} := (\mathbb{T}, \mathbb{W}, \mathfrak{B}) = (\mathbb{N}, \mathbb{U} \times \mathbb{X} \times \mathbb{Y}, \mathfrak{B}^{\mathrm{init}}_{\mathrm{i/s/o}}) \tag{4}$$

with the time axis $\mathbb{T} := \mathbb{N} = \{0, 1, 2, \ldots\}$, the full signal space $\mathbb{W} := \mathbb{U} \times \mathbb{X} \times \mathbb{Y}$, and the *initialised* behaviour

$$\begin{aligned} \mathfrak{B}^{\mathrm{init}}_{\mathrm{i/s/o}} = \{&w \in \mathbb{W}^\mathbb{N} | w = (u, x, y) \text{ s.t. } (1) \\ &\text{and s.t. } x(0) = x_0 \in \mathbb{X}_0\}. \end{aligned}$$

In the sequel, the indices init and $\mathrm{i/s/o}$ of $\mathfrak{B}^{\mathrm{init}}_{\mathrm{i/s/o}}$ will be dropped where it is not ambiguous.

Within this work, we will focus on descriptor systems that are controllable. As in [9, Theorem 8-1.2 and Theorem 2-2.1], we say that a DS $\mathbf{\Sigma} = (E, A, B, C)$ is controllable if

$$\operatorname{rank}(\begin{bmatrix} E & -B \end{bmatrix}) = n \text{ and } \operatorname{rank}(\begin{bmatrix} \sigma E - A & B \end{bmatrix}) = n \, \forall \sigma \in \mathbb{C}.$$

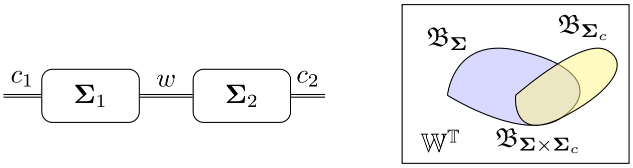
Assumption 1 (Controllable DS)**.** The DS $\mathbf{\Sigma}$ is controllable.

Remark 1. The notion of controllability that is used here is stronger than the 'behavioural' notion of controllability given in [30]. The results in the sequel can be extended to the case where the condition $\operatorname{rank}(\begin{bmatrix} E & -B \end{bmatrix}) = n$ does not hold by using the more general transformation to a driving variable representation given in [26].

B. Interconnections, control, and controller properties

Controller synthesis amounts to synthesising a system $\mathbf{\Sigma}_c$, called a controller, which, after interconnection with a given system $\mathbf{\Sigma}$, restricts the behaviour $\mathfrak{B}$ of $\mathbf{\Sigma}$ to a set of desirable (or controlled) trajectories. In the behavioural framework, control is defined through *interconnections* via variable sharing, similarly to the notion of parallel composition in [3], [23].

Definition 2. *Let $\mathbf{\Sigma}_1 = (\mathbb{T}, \mathbb{C}_1 \times \mathbb{W}, \mathfrak{B}_1)$ and $\mathbf{\Sigma}_2 = (\mathbb{T}, \mathbb{C}_2 \times \mathbb{W}, \mathfrak{B}_2)$ be two dynamical systems. Then the* interconnection *of $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ over $\mathbb{W}$, denoted by $\mathbf{\Sigma} = \mathbf{\Sigma}_1 \times_w \mathbf{\Sigma}_2$ with the shared variable $w \in \mathbb{W}$, yields the dynamical system $\mathbf{\Sigma} = (\mathbb{T}, \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{W}, \mathfrak{B})$ with $\mathfrak{B} = \{(c_1, c_2, w) : \mathbb{T} \to \mathbb{C}_1 \times \mathbb{C}_2 \times \mathbb{W} \mid (c_1, w) \in \mathfrak{B}_1, (c_2, w) \in \mathfrak{B}_2\}$.*


c_1
$\mathbf{\Sigma}_1$
w
$\mathbf{\Sigma}_2$
c_2
$\mathfrak{B}_{\mathbf{\Sigma}}$
$\mathfrak{B}_{\mathbf{\Sigma}_c}$
$\mathbb{W}^{\mathbb{T}}$
$\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}$

(a) The interconnected system $\mathbf{\Sigma}$ obtained via the shared variables w in $\mathbb{W}$ between dynamical systems $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ with signal spaces $\mathbb{C}_1 \times \mathbb{W}$ and $\mathbb{C}_2 \times \mathbb{W}$.

(b) The controlled behaviour $\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c} = \mathfrak{B}_{\mathbf{\Sigma}} \cap \mathfrak{B}_{\mathbf{\Sigma}_c}$ is given as the intersection of the behaviours of the dynamical system $\mathbf{\Sigma}$ and its controller $\mathbf{\Sigma}_c$.

Fig. 1: The left figure (a) portrays the general interconnection of two dynamical systems. Figure (b) depicts the specific case of behavioural intersection between a system and its controller.

This *partial interconnection* is also shown in Fig 1a. If both $\mathbb{C}_1$ and $\mathbb{C}_2$ are empty, we call $\mathbf{\Sigma} = \mathbf{\Sigma}_1 \times_w \mathbf{\Sigma}_2$ a *full interconnection* and $\mathfrak{B}_{\mathbf{\Sigma}_1 \times \mathbf{\Sigma}_2} = \mathfrak{B}_{\mathbf{\Sigma}_1} \cap \mathfrak{B}_{\mathbf{\Sigma}_2}$ as portrayed in Fig. 1b.

Control design therefore amounts to synthesizing a dynamical system $\mathbf{\Sigma}_c$ that can be interconnected with the given system [29], [30]. This framework of control design is more general than the common setting in which controllers are viewed as causal operators. Obviously, of specific interests are controllers that lend themselves to a physical implementation. Firstly, we define a well-posed controller $\mathbf{\Sigma}_c$ for $\mathbf{\Sigma}$ as follows.

Definition 3 (Well-posed controller)**.** Consider a dynamical system $\mathbf{\Sigma} = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, with initialised behaviour as defined in (4). We say that a system $\mathbf{\Sigma}_c = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_c)$ is a *well-posed* controller for $\mathbf{\Sigma}$ if for every initial state $x_0 \in \mathbb{X}_0$ of $\mathbf{\Sigma}$, there exists a unique continuation in $\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}$. Denote by $\mathfrak{C}(\mathbf{\Sigma})$ the collection of all well-posed controllers for $\mathbf{\Sigma}$.

As formalised by the second condition, a well-posed controller is required to accept any initial state of the system. That is, for any initial state of $\mathbf{\Sigma}$, there should exist a continuation in $\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}$. This restricts the choice of controllers to those for which $\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c} := \mathfrak{B}_{\mathbf{\Sigma}} \cap \mathfrak{B}_{\mathbf{\Sigma}_c} \neq \{\emptyset\}$, i.e., the composed system is non-blocking as in [23].

Lemma 1 (Well-posed controller)**.** *For a system $\mathbf{\Sigma}$ as in* (1)*, consider a controller $\mathbf{\Sigma}_c$, which is a DS, and has dynamics given as*

$$E_c x(t+1) = A_c x(t) + B_c u(t), \tag{5}$$

with $E_c, A_c \in \mathbb{R}^{n_c \times n}$ and $B_c \in \mathbb{R}^{n_c \times p}$. Let the controller be initialised with $x(0)$ and, moreover, suppose that the controller shares the variables u and x with the system $\mathbf{\Sigma}$. That is, $w = (u, x)$. Then the interconnected system $\mathbf{\Sigma} \times_w \mathbf{\Sigma}_c$ has the state evolution described by

$$\begin{bmatrix} E \\ E_c \end{bmatrix} x(t+1) = \begin{bmatrix} A \\ A_c \end{bmatrix} x(t) + \begin{bmatrix} B \\ B_c \end{bmatrix} u(t), \tag{6}$$

and can be rewritten as

$$\begin{bmatrix} E & -B \\ E_c & -B_c \end{bmatrix} \begin{bmatrix} x(t+1) \\ u(t) \end{bmatrix} = \begin{bmatrix} A \\ A_c \end{bmatrix} x(t). \tag{7}$$

If for any $x(t) \in \mathbb{X}$, there exists a unique pair $(x(t+1), u(t))$ such that (7) holds, then for any initial state $x_0 \in \mathbb{X}_0$ of $\mathbf{\Sigma}$ there exists a unique continuation in the controlled behaviour and $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$. Specifically, the controller is well-posed, i.e., $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ if and only if

$$\operatorname{rank}\left(\begin{bmatrix} E & B \\ E_c & B_c \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} E & B & A \\ E_c & B_c & A_c \end{bmatrix}\right) = n+p. \tag{8}$$

Proof. Starting from a well-posed controller $\mathbf{\Sigma}_c$, we know that the augmented matrix on the left of (7) has full column rank, thus it has a left inverse. Multiplying this left inverse by the left on both sides of (7), shows that the controlled system can be represented by an autonomous system in standard state space form. Therefore it has a unique continuation. □

More stringent specific controller properties are related to how the controller can be implemented on the system. There are several types of these compatibility conditions both for general nonlinear dynamical systems [16] and conditions specifically for linear dynamical systems [28]. Within this work, we will use the notion of regular interconnections defined specifically for linear dynamical systems. The class of linear time-invariant dynamical systems that we consider here admit difference equation representations of the generic form $R(q)w(t) = 0$ for polynomial operators R in the shift q [30]. For any such representation, the signal space can be written as $\mathbb{W} = \mathbb{U} + \mathbb{Y}$ where $\mathbb{U}$ denotes the input space of free (unconstrained) variables and $\mathbb{Y}$ the output space of constrained variables. It is shown in [28] that such a decomposition is not unique but the dimension $p(\mathbf{\Sigma}) := \dim(\mathbb{Y})$ of the output space is. Precisely, $p(\mathbf{\Sigma}) = rank(R)$ whenever $\mathfrak{B} = \ker R(q)$. A la [28], we define a regular interconnection as

Definition 4 (Regular interconnection [28])**.** Let $\mathbf{\Sigma}, \mathbf{\Sigma}'$ be linear dynamical systems. We call $\mathbf{\Sigma} \times \mathbf{\Sigma}'$ a regular interconnection if $p(\mathbf{\Sigma} \times \mathbf{\Sigma}') = p(\mathbf{\Sigma}) + p(\mathbf{\Sigma}')$.

This notion of regularity also implies weak uniform compliance [16], which is a notion applicable to the more general set of nonlinear systems. Physically, weakly uniformly compliant systems can be interconnected to each other by first preparing them [16]. Based on Assumption 1, we can use the following proposition of [28].

Proposition 2. *Assume that the linear dynamical system $\mathbf{\Sigma}$ is controllable. Let the linear dynamical system $\mathbf{\Sigma}''$ be a subsystem of $\mathbf{\Sigma}$, i.e., $\mathfrak{B}_{\mathbf{\Sigma}''} \subset \mathfrak{B}_{\mathbf{\Sigma}}$. Then there exists a system $\mathbf{\Sigma}'$ such that $\mathbf{\Sigma} \times \mathbf{\Sigma}' = \mathbf{\Sigma}''$ and such that this interconnection is regular.*

Based on this proposition, we can connect the notion of well-posed controllers to the existence of controllers that yield a regular interconnection.

Corollary. *If Assumption 1 holds, then $\mathbf{\Sigma}$ is a linear controllable system and the existence of a linear well-posed controller $\mathbf{\Sigma}_c$ implies the existence of a controller $\tilde{\mathbf{\Sigma}}_c$ that yields a regular interconnection such that $\mathbf{\Sigma} \times \mathbf{\Sigma}_c = \mathbf{\Sigma} \times \tilde{\mathbf{\Sigma}}_c$.*

Proof. The interconnection of the linear well-posed controller $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ with the linear DS system $\mathbf{\Sigma}$ is given as $\mathbf{\Sigma}_c \times \mathbf{\Sigma}$. This interconnection is a subsystem of $\mathfrak{B}_{\mathbf{\Sigma}}$, that is, it has the property that $\mathfrak{B}_{\mathbf{\Sigma}_c \times \mathbf{\Sigma}} \subset \mathfrak{B}_{\mathbf{\Sigma}}$. □

This corollary indicates that it is sufficient to focus on the design of a behavioural subset as this implies already the existence of a controller that can be implemented on the system via a regular interconnection.

Of interest is the design of well-posed controllers subject to specifications over the future output behaviour of the closed-loop controlled system. We thus consider specifications defined over the output space. To analyse the output behaviour, we introduce a projection map: for $\mathfrak{B} \subset (\mathbb{W}_1 \times \mathbb{W}_2)^{\mathbb{T}}$, we denote by $\Pi_{\mathbb{W}_2}$ the projection given as

$$\Pi_{\mathbb{W}_2}(\mathfrak{B}) := \{w_2 \in \mathbb{W}_2^{\mathbb{T}} \mid \exists w_1 \in \mathbb{W}_1^{\mathbb{T}} \text{ s.t. } (w_1, w_2) \in \mathfrak{B}\}.$$

This is depicted in Figure 2 for control synthesis; we focus on finding a controller $\mathbf{\Sigma}_c$ for a given dynamical system $\mathbf{\Sigma}$ such that the output behaviour $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c})$ of the interconnected system satisfies some given specification.

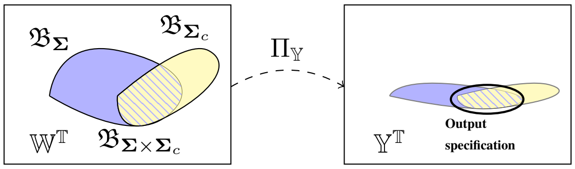

$\mathfrak{B}_{\mathbf{\Sigma}}$
$\mathfrak{B}_{\mathbf{\Sigma}_c}$
$\Pi_{\mathbb{Y}}$
$\mathbb{W}^{\mathbb{T}}$
$\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}$
$\mathbb{Y}^{\mathbb{T}}$
Output
specification

Fig. 2: Depiction of a controller designed to satisfy requirement on the output behaviour: the set with the output specifications is given on the right.

Remark 2. For transition systems that are classically employed in formal methods [3], [4], [23] discuss the notions *input words*, *trajectories (or runs, paths)*, and *output words or traces*: these notions can be connected to the notions of input sequences, state trajectories and output trajectories of dynamical and control systems. From this perspective, the behaviour of a dynamical systems combines these notions, that is the set $\mathbb{W}^{\mathbb{T}}$ of a dynamical system $\mathbf{\Sigma}$ with $\mathbb{W} := \mathbb{U} \times \mathbb{X} \times \mathbb{Y}$ can also regarded as the collection of input words, runs and output words. The projection map $\Pi_{\mathbb{Y}}$ discussed in Fig. 2, defining the output behaviour of a dynamical system, performs a mapping that is similar to the derivation of the *language* [4] of a transition system $\mathbf{T}$. More precisely, the set of all output words generated by all runs starting at $x_0 \in \mathbb{X}_0$ is called the language of $\mathbf{T}$ originating at x_0, and is denoted by $\mathcal{L}_{\mathbf{T}}(x_0)$. The language of $\mathbf{T}$ originating from the initial set $\mathbb{X}_0$ is then $\mathcal{L}_{\mathbf{T}}(\mathbb{X}_0) = \cup_{x_0 \in \mathbb{X}_0} \mathcal{L}_{\mathbf{T}}(x_0)$. As per Fig. 2, it is required that the controlled output behaviour satisfies a given output specification: this is a requirement that is similar to that of *language inclusion* in formal methods, which focuses on specifications expressed in temporal logic [8], e.g. LTL [4] formulae. We further elucidate the connection between output specification satisfaction in Fig. 2 and language inclusion for LTL with the following simple example.

Example 3. Consider the dynamical system $\mathbf{\Sigma} = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ over the time axis $\mathbb{T} := \mathbb{N} = \{0, 1, 2, \ldots\}$, endowed with a full signal space $\mathbb{W} := \mathbb{U} \times \mathbb{X} \times \mathbb{Y}$, and the *initialised* behaviour

$$\mathfrak{B}^{\text{init}}_{\text{i/s/o}} = \{w \in \mathbb{W}^{\mathbb{N}} \mid w = (u, x, y) \text{ s.t. (9)} \\ \text{and s.t. } x(0) = x_0 \in \mathbb{X}_0\},$$

where $\mathbb{X}_0 = [-1, 1]$ and

$$\begin{aligned} x(t+1) &= 2x(t) + u(t); \\ y(t) &= x(t), \quad x(0) \in \mathbb{X}_0. \end{aligned} \tag{9}$$

We are interested in designing a feedback controller $\mathbf{\Sigma}_c : u = kx$, such that the controlled output behaviour $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c})$ satisfies the following specification: "the output behaviour $y(t)$ remains in the region $\mathbb{X}_0$ at any time in the future." This specification can be expressed as finding the control policy such that $\mathcal{L}_{\mathbf{T}}(\mathbb{X}_0) \subseteq \mathcal{L}_{\phi}$, where $\mathcal{L}_{\phi}$ denotes the set of words associated to the LTL formula $\phi := \Box\mathbb{X}_0$ (the "always" operator $\Box$ denotes satisfaction at all future times). This goal is achieved by $\mathbf{\Sigma}_c := \{u = kx \mid -3 \leq k \leq -1\}$.

C. Problem statement: Control refinement for DS

Let us refer to the original DS that represents the real physical system as the *concrete* DS. It is for this system that we would like to develop a well-posed controller (c.f. Def. 3) such that the set of allowed output behaviours (derived, say, from a given specification) includes the output behaviour of this system.

Let DS $\mathbf{\Sigma}_a$ be an abstraction of $\mathbf{\Sigma}$, that is, $\mathbf{\Sigma}_a$ is a descriptor system that shares the same output space $\mathbb{Y}$ as $\mathbf{\Sigma}$, and in general $\mathbf{\Sigma}_a$ is a simpler version of $\mathbf{\Sigma}$ in which some part of the dynamics is abstracted away. More precisely, $\mathbf{\Sigma}_a$ has dynamics given as (E_a, A_a, B_a, C_a) and initialised with $\mathbb{X}_{a0}$:

$$\begin{aligned} E_a x_a(t+1) &= A_a x_a(t) + B_a u_a(t); \\ y_a(t) &= C_a x_a(t), \quad x_a(0) \in \mathbb{X}_{a0}, \end{aligned} \tag{10}$$

where $x_a(t) \in \mathbb{X}_a \subseteq \mathbb{R}^m, u_a(t) \in \mathbb{U}_a \subseteq \mathbb{R}^q, y_a(t) \in \mathbb{Y}_a \subseteq \mathbb{R}^k$ with $\mathbb{Y}_a = \mathbb{Y}$. We assume that the synthesis of a well-posed controller $\mathbf{\Sigma}_{c_a}$ for $\mathbf{\Sigma}_a$ is easier than for $\mathbf{\Sigma}$, because it is of the same or of a lower dimension than the concrete DS system $\mathbf{\Sigma}$, i.e., $m \leq n$. We refer to this simpler system $\mathbf{\Sigma}_a$ as the *abstract* DS. Similarly, the input/output behaviour of the abstract DS $\mathbf{\Sigma}_a$ is derived as

$$\mathfrak{B}_{\mathbf{\Sigma}_a} := \{(u_a, y_a) \in (\mathbb{U}_a \times \mathbb{Y}_a)^{\mathbb{N}} \mid \exists x_a \in \mathbb{X}_a^{\mathbb{N}} \\ \text{s.t. } (u_a, x_a, y_a) \text{ satisfies (10)}\}.$$

The controlled abstract system $\mathbf{\Sigma}_a \times_{w_a} \mathbf{\Sigma}_{c_a}$ is the interconnected system with the shared variables $w_a = (u_a, x_a)$.

Given a well-posed controller $\mathbf{\Sigma}_{c_a}$ for the abstract DS $\mathbf{\Sigma}_a$, we are interested in the existence and construction of a well-posed controller $\mathbf{\Sigma}_c$ for $\mathbf{\Sigma}$ such that the output behaviour of the two controlled systems is exactly the same for *exact refinement*, or the distance between the outputs is bounded by ε for *approximate refinement*. This setup leads to the notion of control refinement, which is depicted in Fig. 3 and formalised next.

Definition 5 (Control refinement)**.** Let $\mathbf{\Sigma}_a$ and $\mathbf{\Sigma}$ be the abstract and concrete DS, respectively. We say that controller $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ refines the controller $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$ **exactly** if

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}),$$

and ϵ**-approximately** if

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}})).$$

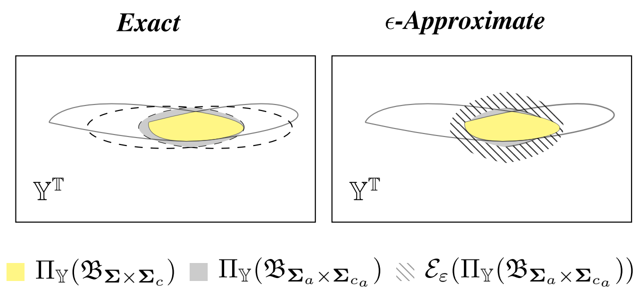

Exact
ϵ-Approximate
$\mathbb{Y}^{\mathbb{T}}$
$\mathbb{Y}^{\mathbb{T}}$
$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c})$
$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}})$
$\mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}))$

Fig. 3: Depiction of the two types of control refinements. The dashed lines encircle the behavioural sets of the abstract DS and its controller. The solid lines define the respective original behavioural sets. When interconnected, the sets $\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}$ and $\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}$ are obtained. The left depicts the exact relation of these behavioural sets in the output signal space. The right illustration gives the ϵ-approximate version.

Recall that $\mathcal{E}_\varepsilon$ is the point-wise ϵ expansion of the signals. Next, the exact control refinement problem is formalised.

Problem 1 (Exact control refinement)**.** Under what conditions on DS systems $\mathbf{\Sigma}_a$ and $\mathbf{\Sigma}$ does there exist a well-posed or regular controller $\mathbf{\Sigma}_c$ for every $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$ that results in an **exact control refinement**, as per Definition 5?

In a similar fashion, we are also interested in the approximate notion, as it is expected that it allows for more freedom in the abstraction and the controller design.

Problem 2 (Approximate control refinement)**.** Under what conditions on DS systems $\mathbf{\Sigma}_a$ and $\mathbf{\Sigma}$ does there exist a well-posed or a regular controller $\mathbf{\Sigma}_c$ for every $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$ that results in an ϵ**-approximate control refinement**, as defined in Def.5?

Problems 1 and 2 are very similar to standard (approximate) control refinement problems [13], [23] with the exception that the compatibility conditions have been added.

We approach these problems by introducing notions of (approximate) similarity between descriptor systems and show that they allow us to prove the existence of solutions to Problem 1 and Problem 2.

III. Similarity relations between DS

To quantify whether two systems are similar to each other we can use either the exact or the approximate notions of simulation and bi-simulation [2]. Originally defined in formal verification for finite-state transition systems [3] and later employed for control systems [23], we will recall these notions and taylor them to descriptor systems, also thanks to the use of behavioural theory. For pairs of DS $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ (cf. Def. 1) with equal output space $\mathbb{Y}_1 = \mathbb{Y}_2 = \mathbb{Y}$, a simulation relation is defined as follows.

Definition 6 (Exact simulation)**.** Let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be two DS with corresponding dynamics (E_1, A_1, B_1, C_1) and (E_2, A_2, B_2, C_2) over state spaces $\mathbb{X}_1$ and $\mathbb{X}_2$. A relation $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ is called a *simulation relation from* $\mathbf{\Sigma}_1$ *to* $\mathbf{\Sigma}_2$, if $\forall (x_1, x_2) \in \mathcal{R}$,

1) for all pairs of control input and next states $(u_1, x_1^+) \in \mathbb{U}_1 \times \mathbb{X}_1$ subject to
$$E_1 x_1^+ = A_1 x_1 + B_1 u_1 \tag{11}$$
there exists a control input and next state pair $(u_2, x_2^+) \in \mathbb{U}_2 \times \mathbb{X}_2$ subject to
$$E_2 x_2^+ = A_2 x_2 + B_2 u_2 \tag{12}$$
such that $(x_1^+, x_2^+) \in \mathcal{R}$, and
2) it holds that $C_1 x_1 = C_2 x_2$.

We say that $\mathbf{\Sigma}_1$ *is simulated by* $\mathbf{\Sigma}_2$, denoted by $\mathbf{\Sigma}_1 \preceq \mathbf{\Sigma}_2$, if there exists a simulation relation $\mathcal{R}$ from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$ and if in addition $\forall x_{10} \in \mathbb{X}_{10}, \exists x_{20} \in \mathbb{X}_{20}$ such that $(x_{10}, x_{20}) \in \mathcal{R}$.

We call $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ a *bisimulation relation* between $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ if $\mathcal{R}$ is a simulation relation from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$ and its inverse $\mathcal{R}^{-1} \subseteq \mathbb{X}_2 \times \mathbb{X}_1$ is a simulation relation from $\mathbf{\Sigma}_2$ to $\mathbf{\Sigma}_1$. We say that $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are bisimilar, denoted by $\mathbf{\Sigma}_1 \cong \mathbf{\Sigma}_2$, if $\mathbf{\Sigma}_1 \preceq \mathbf{\Sigma}_2$ w.r.t. $\mathcal{R}$ and $\mathbf{\Sigma}_2 \preceq \mathbf{\Sigma}_1$ w.r.t. $\mathcal{R}^{-1}$.

The notion of approximate simulation relation is obtained by relaxing the equality of the output behaviour. Instead of identical behaviours, for an approximate simulation relation we require that the distance between these output behaviours remains bounded.

Definition 7 (Approximate simulation)**.** Let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be two DS with corresponding dynamics (E_1, A_1, B_1, C_1) and (E_2, A_2, B_2, C_2) over state spaces $\mathbb{X}_1$ and $\mathbb{X}_2$. A relation $\mathcal{R}_\varepsilon \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ is called an *approximate simulation relation from* $\mathbf{\Sigma}_1$ *to* $\mathbf{\Sigma}_2$, if $\forall (x_1, x_2) \in \mathcal{R}_\varepsilon$,

1) for all $(u_1, x_1^+) \in \mathbb{U}_1 \times \mathbb{X}_1$ subject to Eq. (11) there exists $(u_2, x_2^+) \in \mathbb{U}_2 \times \mathbb{X}_2$ subject to Eq. (12) such that $(x_1^+, x_2^+) \in \mathcal{R}_\varepsilon$, and
2) we have $d(C_1 x_1, C_2 x_2) \leq \varepsilon$.

We say that $\mathbf{\Sigma}_2$ approximately simulates $\mathbf{\Sigma}_1$, denoted by $\mathbf{\Sigma}_1 \preceq_\varepsilon \mathbf{\Sigma}_2$, if there exists an approximate simulation relation $\mathcal{R}_\varepsilon$ from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$ and if in addition $\forall x_{10} \in \mathbb{X}_{10}, \exists x_{20} \in \mathbb{X}_{20}$ such that $(x_{10}, x_{20}) \in \mathcal{R}_\varepsilon$.

We now consider the following example to get more insights of the simulation relations between DS and their behaviours.

Example 4. Consider a concrete DS $\mathbf{\Sigma}$ with dynamics (E, A, B, C), defined as

$$E = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{smallmatrix}\right], A = \left[\begin{smallmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right], B = \left[\begin{smallmatrix} 1 \\ 2 \\ 0 \end{smallmatrix}\right], C = \left[\begin{smallmatrix} 0.2 \\ 0.5 \\ 1 \end{smallmatrix}\right]^T$$

with $\mathbb{X}_0 \subseteq \mathbb{X}$ and $\mathbb{X}, \mathbb{U}, \mathbb{Y}$ are respectively subsets of $\mathbb{R}^3, \mathbb{R}, \mathbb{R}$. The DS Σ is not a minimal realisation because it is observable but not reachable, see [6], [9] for details on observability and reachability. Based on the Silverman-Ho algorithm [9], we choose an abstract DS $\mathbf{\Sigma}_a$ with dynamics (E_a, A_a, B_a, C_a) that is the minimal realization of $\mathbf{\Sigma}$ and

$$E_a = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right], A_a = \left[\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right], B_a = \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right], C_a = \left[\begin{smallmatrix} 0.2 \\ 1 \end{smallmatrix}\right]^T.$$

Similarly, $\mathbb{X}_{a0} \subseteq \mathbb{X}_a$ and $\mathbb{X}_a, \mathbb{U}_a, \mathbb{Y}$ are respectively subsets of $\mathbb{R}^2, \mathbb{R}, \mathbb{R}$. Subsequently,

$$\mathcal{R} := \{(x_a, x) \mid x = \mathcal{H}x_a, x_a \in \mathbb{X}_a, x \in \mathbb{X}\}$$

is a bisimulation relation between Σ_a and Σ, where

$$\mathcal{H} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{smallmatrix}\right].$$

Then, we consider the two requirements of bisimulation relations. For any $(x_a, x) \in \mathcal{R}$, we have $Cx = C\mathcal{H}x_a = C_a x_a$. For any $(x_a, x) \in \mathcal{R}$ with $x_a = (x_{a1}, x_{a2})^T$ and $x = (x_1, x_2, 0)^T$ subject to $x_1 = x_{a1}, x_2 = 2x_{a2}$, it holds that for the state evolution $x_a \xrightarrow{u_a}_a x_a^+$ in $\mathbf{\Sigma}_a$ the input is such that $u_a = -x_{a2}$. Still the next state $x_a^+ = (-x_{a1} - x_{a2}, x_{a2}^+)$ has a free variable x_{a2}^+. For the original system $\mathbf{\Sigma}$, the next state $x^+ = (-x_1 - 0.5x_2, x_2^+)$ can be chosen as $x_2^+ = 2x_{a2}^+$, such that $(x_a^+, x^+) \in \mathcal{R}$. Thereby the relation $\mathcal{R}$ is a simulation relation, by proving that the reverse also holds, we can show that the relation $\mathcal{R}$ is a bisimulation.

In addition, if all $x_0 \in \mathbb{X}_0$ can be mapped to $(x_{10}, x_{20}, 0)^T$, then there always exists $x_{a0} = (x_{a10}, x_{a20})^T = (x_{10}, 0.5x_{20})^T \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$. Conversely, for any $x_{a0} \in \mathbb{X}_{a0}$, there exists $x_0 = \mathcal{H}x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$. Therefore, we can conclude that $\mathbf{\Sigma}_a \cong \mathbf{\Sigma}$.

The given notions of (bi)simulation relations and approximate simulation relations are a specific case of those defined for metric transition systems [3], [23]. As such, key properties are preserved. This includes the transitivity of these relations, as given next, that are key for the results in this work.

Proposition 3 (Transitivity)**.** *Let $\mathcal{R}_{\varepsilon_1}$ be an approximate simulation relation from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$ and $\mathcal{R}_{\varepsilon_2}$ be an approximate simulation relation from $\mathbf{\Sigma}_2$ to $\mathbf{\Sigma}_3$. In addition, $\forall x_{10} \in \mathbb{X}_{10}, \exists x_{20} \in \mathbb{X}_{20}$ s.t. $(x_{10}, x_{20}) \in \mathcal{R}_{\varepsilon_1}$ and $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{30} \in \mathbb{X}_{30}$ s.t. $(x_{20}, x_{30}) \in \mathcal{R}_{\varepsilon_2}$. Then, we can conclude that*

$$\begin{aligned} \mathcal{R}_{\varepsilon_1+\varepsilon_2} := \mathcal{R}_{\varepsilon_1} \circ \mathcal{R}_{\varepsilon_2} = \{&(x_1, x_3) \mid \exists x_2 \\ &\text{s.t. } (x_1, x_2) \in \mathcal{R}_{\varepsilon_1} \wedge (x_2, x_3) \in \mathcal{R}_{\varepsilon_2}\} \end{aligned}$$

is an approximate simulation relation from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_3$, and in addition $\forall x_{10} \in \mathbb{X}_{10}, \exists x_{30} \in \mathbb{X}_{30}$ s.t. $(x_{10}, x_{30}) \in \mathcal{R}_{\varepsilon_1+\varepsilon_2}$.

By setting ε_1 and ε_2 to 0, we also get that transitivity holds for exact simulation relations.

Simulation relations imply properties of the output behaviours of the two systems: this follows from [3], [23] and is formalised next.

Proposition 4. *Let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be two DS with similarity relations as defined in Definitions 6 and 7. Then, the following implications hold:*

$$\begin{aligned} &\mathbf{\Sigma}_1 \preceq \mathbf{\Sigma}_2 \Longrightarrow \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2}), \\ &\mathbf{\Sigma}_1 \cong \mathbf{\Sigma}_2 \Longrightarrow \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2}), \\ &\mathbf{\Sigma}_1 \preceq_{\varepsilon} \mathbf{\Sigma}_2 \Longrightarrow \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1}) \subseteq \mathcal{E}_{\varepsilon}(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2})). \end{aligned}$$

Proof. Based on [3], [23], we can infer the first two implications. The last implication follows from the definition of approximate simulation relation. For every signal $y \in \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2})$, there exists a state signal and input signal x, u such that $(x_1, u_1) \in \mathfrak{B}^{\text{init}}_{\text{i/s/o},1}$. Since $\mathbf{\Sigma}_1 \preceq_{\varepsilon} \mathbf{\Sigma}_2$, we know that for $x_1(0) \in \mathbb{X}_{10}$ there exists $x_2(0) \in \mathbb{X}_{20}$ with $(x_1(0), x_2(0)) \in \mathcal{R}$. Moreover for the state-input signal (x_1, u_1) there exists an input signal u_2 such that $(x_1(t), x_2(t)) \in \mathcal{R}$ for all $t \geq 0$. The corresponding output signal y_2 satisfies $d(y_1(t), y_2(t)) \leq \epsilon$ for all $t \geq 0$. Therefore, it also holds that $y_1(t) \in \mathcal{E}_{\varepsilon}(y_2)$. By extension this also proves the third implication. □

Remark 5. The implications in Proposition 4 are unidirectional, that is $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2})$ does not imply that $\mathbf{\Sigma}_1 \preceq \mathbf{\Sigma}_2$. Indeed, consider the two non-singular DS's

$$\mathbf{\Sigma}_a : \begin{cases} x_a(t+1) & = u_a(t) \\ y_a(t) & = x_a(t) \end{cases}$$

$$\mathbf{\Sigma}_b : \begin{cases} x_b(t+1) & = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right] x_b(t) + \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right] u_b(t) \\ y_b(t) & = \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right] x_b(t) \end{cases}$$

with $u_a(t), u_b(t) \in \mathbb{R}$ and $x_b(t) = \begin{bmatrix} x_{b1}(t) & x_{b2}(t) \end{bmatrix}^T$. $\mathbf{\Sigma}_a$ is initialised with $x_a(0) \in \mathbb{R}$ while $\mathbf{\Sigma}_b$ is initialised with $x_b(0) \in \mathbb{R}^2$. For any initial state $x_a(0)$ of $\mathbf{\Sigma}_a$ and the input sequence $u_a(t)$, we can produce the same trajectory in $\mathbf{\Sigma}_b$ by choosing $x_{b1}(0) = x_a(0), x_{b2}(0) = u_a(0)$ and $u_b(t) = u_a(t+1)$. Therefore, we have that $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_b})$ since for every output trajectory in $\mathbf{\Sigma}_a$, there exists the same output trajectory in $\mathbf{\Sigma}_b$. But it does not hold that $\mathbf{\Sigma}_a \preceq \mathbf{\Sigma}_b$. This is because the pairs of control input and next states $(u_a(t), x_a(t+1))$ in $\mathbf{\Sigma}_a$ cannot be matched by the pairs of control input and next states $(u_b(t), x_b(t+1))$ in $\mathbf{\Sigma}_b$ since neither the initialisation $x_{b2}(0) = u_a(0)$ nor the anticipating input $u_b(t) = u_a(t+1)$ in $\mathbf{\Sigma}_b$ is guaranteed.

Simulation relations are key concepts for controller design over deterministic systems [10], [13]:[1] in this work, we apply them to discrete-time linear DS and show under what conditions they solve the control refinement problems 1 and 2.

IV. Descriptor and driving variable systems

Since it is difficult to control and analyse a DS directly, we develop a transformation to a system representation that is in the non-singular DS form and is driven by an auxiliary input. We refer to this non-singular DS as the driving variable (DV) system [24]. We investigate whether the DS and the obtained DV system are bisimilar and therefore behaviourally

[1] A system is called *deterministic* if for any state $x \in \mathbb{X}$ and any input $u \in \mathbb{U}$, $x \xrightarrow{u} x^+$ and $x \xrightarrow{u} x^{+\prime}$ implies $x^+ = x^{+\prime}$. A system is called *nondeterministic* if it is not deterministic.

equivalent. Let us first introduce by a simple example the apparent non-determinism or anti-causality in the DS. Later we show the connections between a DS and its corresponding DV system.

Example 6. *Consider the DS with dynamics* (E, A, B, C) *defined as*

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 0.2 \\ 0.5 \end{bmatrix}^T, \tag{13}$$

and $x(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix}^T$. *In this case, we have that* $u(t) + x_3(t) = 0$ *holds. The state trajectories of* (13) *can be found as follows:*

- *for a given input sequence* $u : \mathbb{T} \to \mathbb{U}$, $x_2(t) = -u(t) - u(t+1)$, *and thus we can use this anti-causal relation of the DS to find the corresponding state trajectories;*
- *alternatively we can allow the next state* $x_2(t+1)$ *to be freely chosen, for arbitrary state* $x_2(t)$, *Equations* (13) *impose a relation between the input sequence* $u(t)$ *and the state* $x_3(t)$, *that is* $u(t) + x_3(t) = 0$. *The input* $u(t)$ *here is actually free.*

We embrace the latter, non-deterministic interpretation.

This non-determinism can be characterised by introducing an auxiliary driving input of a so-called DV system. As in Equation (7) of Example 1, we reorganise the state evolution of (1). For simplicity, we omit the time index in $x(t)$ and $u(t)$ and denote $x(t+1)$ as x^+, so that

$$M \begin{bmatrix} x^+ \\ u \end{bmatrix} = Ax, \tag{14}$$

where $M = \begin{bmatrix} E & -B \end{bmatrix} \in \mathbb{R}^{n \times (n+p)}$. If there exists a solution for x, then the solution pairs (u, x^+) are non-unique, due to the non-determinism related to x^+. Based on Assumption 1, M has full row rank, therefore it has a right inverse. This assumption always holds when the DS is reachable (Definition 2-1.1 [9]). In that case we can characterise the non-determinism as follows. Let M_{right}^{-1} be a right inverse of M such that $MM_{\text{right}}^{-1} = I$ and N be a matrix such that $\operatorname{im} N = \ker M$ and $N^T N = I$. Then all pairs (u, x^+) that are compatible with state x in (14) are parametrised as

$$\begin{bmatrix} x^+ \\ u \end{bmatrix} = M_{\text{right}}^{-1} Ax + Ns, \tag{15}$$

where s is a free variable. We now claim that all transitions (x, u, x^+) that satisfy (15) for some variable s satisfy (14). To see this, multiply M on both sides of (15) to obtain (14). Now assume that there exists a tuple (x, u, x^+) satisfying (14) that does not satisfy (15). Then there exists an s and a vector $z \neq 0$ that is not an element of the kernel of M and such that the right side of (15) becomes $M_{\text{right}}^{-1} Ax + Ns + z$. Multiplying again with M, we infer that there is an additional non-zero term Mz and that (14) cannot hold. In conclusion, any transition of (14) is also a transition of (15) and vice versa.

Example 7. *[Example 6, continued] For the DS of Example 6, the related DV system is*

$$\begin{aligned} x(t+1) &= \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} s(t), \\ u(t) &= \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} x(t), \\ y(t) &= \begin{bmatrix} 0 & 0.2 & 0.5 \end{bmatrix} x(t). \end{aligned} \tag{16}$$

As indicated by (16), *the input* $u(t)$ *is a function of the state trajectory. The non-determinism of* $x_2(t+1)$ *is characterised by* $-s(t)$, *for which the auxiliary input* s *can be freely selected.*

Let us now formalise the notion of a driving variable system, which is denoted as $\mathbf{\Sigma}_{\text{DV}}$. We associate a driving variable representation to any given DS (1) by defining a tuple (A_d, B_d, C_u, D_u, C), with

$$\begin{bmatrix} A_d \\ C_u \end{bmatrix} := M_{\text{right}}^{-1} A, \quad \begin{bmatrix} B_d \\ D_u \end{bmatrix} := N. \tag{17}$$

For any given DS, this tuple defines the *driving variable system* $\mathbf{\Sigma}_{\text{DV}} = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_{\mathbf{\Sigma}_{\text{DV}}})$, which maintains the same set of initial states $\mathbb{X}_0$ and has dynamics

$$\begin{aligned} x(t+1) &= A_d x(t) + B_d s(t), \\ u(t) &= C_u x(t) + D_u s(t), \\ y(t) &= C x(t), \quad x(0) \in \mathbb{X}_0, \end{aligned} \tag{18}$$

with $x(t) \in \mathbb{X} \subseteq \mathbb{R}^n, u(t) \in \mathbb{U} \subseteq \mathbb{R}^p, y(t) \in \mathbb{Y} \subseteq \mathbb{R}^k, \mathbb{X}_0 \subseteq \mathbb{X}$. The new variable $s(t)$ is referred to as the *driving variable* which assumes values in $\mathbb{S} \subseteq \mathbb{R}^p$. This yields the initialised behaviour

$$\mathfrak{B}_{\mathbf{\Sigma}_{\text{DV}}} := \{ w \in \mathbb{W}^{\mathbb{T}} \mid w = (u, x, y), \exists s \in \mathbb{S}^{\mathbb{T}} \text{ s.t. (18)} \}.$$

Any concrete DS (1) that is reachable can be rewritten as the corresponding concrete DV system (18).

Conversely, we can develop an algorithm to rewrite the DV system (18) back into a DS $\mathbf{\Sigma}$. In addition, the driving variable s can be expressed by x, x^+ and u. The algorithm is developed based on the singular value decomposition (SVD) of $\begin{bmatrix} B_d^T & D_u^T \end{bmatrix}^T$.

It is easy to see that $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\text{DV}}$ are behaviourally equivalent. Since, as shown in Section 3, bisimilarity always implies output behavioural equivalence, the following proposition proposes the stronger relationship of bisimilarity between a DS and its related DV system.

Theorem 5. *Let the DS* $\mathbf{\Sigma}$ *be given as in* (1) *and let* $\mathbf{\Sigma}_{DV} = (\mathbb{T}, \mathbb{W}, \mathfrak{B}_{\mathbf{\Sigma}_{DV}})$ *be defined as in* (18). *Then:*

(a) $\mathbf{\Sigma}$ *and* $\mathbf{\Sigma}_{DV}$ *are bisimilar, that is,* $\mathbf{\Sigma} \cong \mathbf{\Sigma}_{DV}$,
(b) $\mathbf{\Sigma}$ *and* $\mathbf{\Sigma}_{DV}$ *have equal behaviour, that is,* $\mathfrak{B}_{\mathbf{\Sigma}_{DV}} = \mathfrak{B}_{\mathbf{\Sigma}}$,
(c) $\mathbf{\Sigma}$ *and* $\mathbf{\Sigma}_{DV}$ *have equal output behaviour, that is,* $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}}) = \Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_{DV}})$.

Proof. For the first statement (a), we define the diagonal relation as $\mathcal{I} := \{(x, x) \mid x \in \mathbb{X}\}$. Then $\mathcal{I}$ is a bisimulation relation between $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\text{DV}}$, because by construction their state evolutions can be matched, hence stay in $\mathcal{I}$; and they share the same output map. In addition, since they have the same set of initial states it follows that $\mathbf{\Sigma} \cong \mathbf{\Sigma}_{\text{DV}}$. The matching of the state evolutions can be found in the paragraph following Equation (15).

The second part (b) follows immediately from the derivation of $\mathbf{\Sigma}_{\text{DV}}$, because by construction all the transitions in $\mathbf{\Sigma}$ can be matched by those of $\mathbf{\Sigma}_{\text{DV}}$ and vice versa, in addition, they have the same output map. Hence, they share the same signal space $(\mathbb{U} \times \mathbb{X} \times \mathbb{Y})$ and we can conclude that $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\text{DV}}$ have equal behaviour.

Additionally, we have that (b) implies (c) and via Proposition 4, also that (a) implies (c). □

V. Exact control refinement for descriptor systems

Based on the previous section, we now derive the solution to the exact control refinement goal in Problem 1. For this we exploit the existence of a simulation relation $\mathcal{R}$ from the abstract to the concrete system. More precisely, subject to the assumption that there exists a simulation relation $\mathcal{R}$ from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}$, for which in addition it holds that[2] $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$, we show that for any $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$, there exists a controller $\boldsymbol{\Sigma}_c$ for $\boldsymbol{\Sigma}$ that refines $\boldsymbol{\Sigma}_{c_a}$ such that $\boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma})$ and $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}\times\boldsymbol{\Sigma}_c}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_a\times\boldsymbol{\Sigma}_{c_a}})$. Thus, note that we directly use the simulation relation $\mathcal{R}$ over the state spaces and not the induced preorder notion $\preceq$.

Under Assumption 1, we construct DV systems $\boldsymbol{\Sigma}_{\mathrm{DV}}$ and $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$ for the respective DS systems $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_a$ as a first step. $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$ is the abstract DV system with dynamics $(A_{da}, B_{da}, C_{u_a}, D_{u_a}, C_a)$ defined as

$$\begin{aligned} x_a(t+1) &= A_{da}x_a(t) + B_{da}s_a(t); \\ u_a(t) &= C_{u_a}x_a(t) + D_{u_a}s_a(t); \\ y_a(t) &= C_a x_a(t), \quad x_a(0) \in \mathbb{X}_{a0}, \end{aligned} \tag{19}$$

where $u_a(t) = C_{u_a}x_a(t) + D_{u_a}s_a(t)$ and $x_a(t) \in \mathbb{X}_a \subseteq \mathbb{R}^m, s_a(t) \in \mathbb{S}_a \subseteq \mathbb{R}^q, u_a(t) \in \mathbb{U}_a \subseteq \mathbb{R}^q, y_a(t) \in \mathbb{Y} \subseteq \mathbb{R}^k, \mathbb{X}_{a0} \subseteq \mathbb{X}_a$. Again, $s_a(t)$ is a driving variable. The behaviour of the abstract DV system (19) is denoted by $\mathfrak{B}_{\boldsymbol{\Sigma}_{\mathrm{DV}_a}}$. For DS $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_a$ and their related DV systems $\boldsymbol{\Sigma}_{\mathrm{DV}}$, $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$, we develop the following results on exact control refinement:

(I) The exact control refinement for the DV systems:

$$\begin{aligned} &\forall \boldsymbol{\Sigma}^c_{\mathrm{DV}_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_{\mathrm{DV}_a}), \exists \boldsymbol{\Sigma}^c_{\mathrm{DV}} \in \mathfrak{C}(\boldsymbol{\Sigma}_{\mathrm{DV}}), \text{ s.t.} \\ &\qquad \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{\mathrm{DV}}\times\boldsymbol{\Sigma}^c_{\mathrm{DV}}}\big) \subseteq \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{\mathrm{DV}_a}\times\boldsymbol{\Sigma}^c_{\mathrm{DV}_a}}\big); \end{aligned} \tag{20}$$

(II) The exact control refinement from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$:

$$\begin{aligned} &\forall \boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a), \exists \boldsymbol{\Sigma}^c_{\mathrm{DV}_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_{\mathrm{DV}_a}), \text{ s.t.} \\ &\qquad \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_a\times\boldsymbol{\Sigma}_{c_a}}\big) = \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{\mathrm{DV}_a}\times\boldsymbol{\Sigma}^c_{\mathrm{DV}_a}}\big); \end{aligned} \tag{21}$$

(III) The exact control refinement from $\boldsymbol{\Sigma}_{\mathrm{DV}}$ to $\boldsymbol{\Sigma}$:

$$\begin{aligned} &\forall \boldsymbol{\Sigma}^c_{\mathrm{DV}} \in \mathfrak{C}(\boldsymbol{\Sigma}_{\mathrm{DV}}), \exists \boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma}), \text{ s.t.} \\ &\qquad \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{\mathrm{DV}}\times\boldsymbol{\Sigma}^c_{\mathrm{DV}}}\big) = \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}\times\boldsymbol{\Sigma}_c}\big). \end{aligned} \tag{22}$$

It will be shown that the combination of the elements (I)-(III) also implies the construction of the exact control refinement for the concrete $\boldsymbol{\Sigma}$ and abstract $\boldsymbol{\Sigma}_a$ systems.

A. Exact control refinement for the DV systems

From Theorem 5, we know that $\boldsymbol{\Sigma} \cong \boldsymbol{\Sigma}_{\mathrm{DV}}$ and $\boldsymbol{\Sigma}_a \cong \boldsymbol{\Sigma}_{\mathrm{DV}_a}$ with respective diagonal relations $\mathcal{I} := \{(x,x) | x \in \mathbb{X}\}$ and $\mathcal{I}_a := \{(x_a,x_a) | x_a \in \mathbb{X}_a\}$. Hence as depicted in Fig. 4 and based on the transitivity of simulation relations, we also derive that $\mathcal{R}$ is a simulation relation from $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$ to $\boldsymbol{\Sigma}_{\mathrm{DV}}$.

Since the DV systems $\boldsymbol{\Sigma}_{\mathrm{DV}}$ and $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$ share the same initial states as the respective DS $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_a$, it also holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$.

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\boldsymbol{\Sigma} \arrow[rr, leftrightarrow, "\boldsymbol{\Sigma} \cong \boldsymbol{\Sigma}_{DV} \text{ w.r.t. } \mathcal{I}"] & & \boldsymbol{\Sigma}_{DV} \\
\boldsymbol{\Sigma}_a \arrow[u, "{\begin{array}{l}(x_a,x) \in \mathcal{R} \\ \forall x_0, \exists x_{a0} : \\ \quad (x_{a0},x_0) \in \mathcal{R}\end{array}}"'] \arrow[rr, leftrightarrow, "\boldsymbol{\Sigma}_a \cong \boldsymbol{\Sigma}_{DV_a} \text{ w.r.t. } \mathcal{I}_a"'] & & \boldsymbol{\Sigma}_{DV_a} \arrow[u, "{\begin{array}{l}\mathcal{R} = \mathcal{I}_a \circ \mathcal{R} \circ \mathcal{I} \\ \forall x_0, \exists x_{a0} : \\ \quad (x_{a0},x_0) \in \mathcal{R}\end{array}}"]
\end{tikzcd}$$

Fig. 4: Connection between DS and DV systems for control refinement. As in the definition of the interface (Def. 8), the connection builds on the existence of a relation and an order in the initialisation of the systems.

Since both systems have standard non-singular dynamics, we can follow the procedure in [13] to refine a controller for $\boldsymbol{\Sigma}_{\mathrm{DV}_a}$ to $\boldsymbol{\Sigma}_{\mathrm{DV}}$. We now extend this result to the behavioural framework. Consider the definition of an interface function used for the refinement as illustrated in Fig. 5.

Definition 8. *(Interface[3]). Let $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ be two non-singular DS with respective dynamics (I_1, A_1, B_1, C_1) and (I_2, A_2, B_2, C_2) over state spaces $\mathbb{X}_1$ and $\mathbb{X}_2$. Then $\mathcal{F} : \mathbb{U}_1 \times \mathbb{X}_1 \times \mathbb{X}_2 \mapsto \mathbb{U}_2$ is an interface function with respect to a given relation $\mathcal{R}$, if the following conditions are satisfied:*

1) *for every $(x_1, x_2) \in \mathcal{R}_\varepsilon$ it holds that for all u_1 for $\boldsymbol{\Sigma}_1$ there exists an input u_2 for $\boldsymbol{\Sigma}_2$ such that $(x_1^+, x_2^+) \in \mathcal{R}_\varepsilon$, and*
2) *$\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$ s.t. $(x_{10}, x_{20}) \in \mathcal{R}_\varepsilon$.*

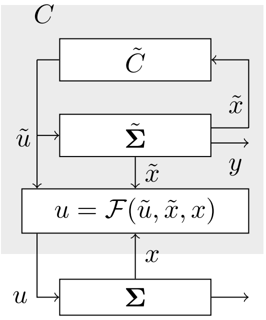

C
$\tilde{C}$
$\tilde{x}$
$\tilde{u}$
$\tilde{\boldsymbol{\Sigma}}$
$\tilde{x}$
y
$u = \mathcal{F}(\tilde{u}, \tilde{x}, x)$
x
u
$\boldsymbol{\Sigma}$

Fig. 5: Control refinement for dynamical models. A controller $\tilde{C}$ for the abstract model $\tilde{\Sigma}$ is refined to the original model Σ.

The definition of an interface function holds for both exact and approximate simulation relations. The existence of an interface function that satisfies the first condition of Definition 8 already follows from the definition of a simulation relation. However, the condition requires a construction of this interface in general. The second condition is instead needed to implement the interface for control refinement.

Proposition 6. *Let $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ be two non-singular DS defined over the same output space $\mathbb{Y}$ with dynamics (I_1, A_1, B_1, C_1) and (I_2, A_2, B_2, C_2), which are initialised with $\mathbb{X}_{10}$ and $\mathbb{X}_{20}$, respectively. If there exists a relation $\mathcal{R} \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ such that*

[2] Notice that the requirement on the initial conditions is in "reverse order" from that needed for $\boldsymbol{\Sigma}_a \preceq \boldsymbol{\Sigma}$.

[3] $\mathcal{F}$ is an interface related to a simulation relation $\mathcal{R}$ by setting $\varepsilon = 0$.

1) $\mathcal{R}$ is a simulation relation from $\boldsymbol{\Sigma}_1$ to $\boldsymbol{\Sigma}_2$ as in Def. 6, and

2) $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$ s.t. $(x_{10}, x_{20}) \in \mathcal{R}$,

then for any controller $\boldsymbol{\Sigma}_{c_1} \in \mathfrak{C}(\boldsymbol{\Sigma}_1)$, there exists a controller $\boldsymbol{\Sigma}_{c_2} \in \mathfrak{C}(\boldsymbol{\Sigma}_2)$ that is an exact control refinement for $\boldsymbol{\Sigma}_{c_1}$ as defined in Def. 5, i.e,

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}_{c_1}}).$$

The proof is actually constructive in the design of the controller $\boldsymbol{\Sigma}_{c_2}$, as it gives a controller that achieves exact or approximate control refinement for $\boldsymbol{\Sigma}_{c_1}$.

Proof. Since $\mathcal{R}$ is a simulation relation from $\boldsymbol{\Sigma}_1$ to $\boldsymbol{\Sigma}_2$, there exists an interface function $\mathcal{F} : \mathbb{U}_1 \times \mathbb{X}_1 \times \mathbb{X}_2 \to \mathbb{U}_2$ related to $\mathcal{R}$ as given in condition (1) of Definition 8, cf [13], [23]. Additionally, due to (2) there exists a map, $\mathcal{F}_0 : \mathbb{X}_{20} \to \mathbb{X}_{10}$ such that for all $x_{20} \in \mathbb{X}_{20}$ it holds that $(\mathcal{F}_0(x_{20}), x_{20}) \in \mathcal{R}$. Next, we *construct the controller* $\boldsymbol{\Sigma}_{c_2}$ that achieves exact control refinement for $\boldsymbol{\Sigma}_{c_1}$ as

$$\boldsymbol{\Sigma}_{c_2} := (\boldsymbol{\Sigma}_1 \times_{w_1} \boldsymbol{\Sigma}_{c_1}) \times_{w_1} \boldsymbol{\Sigma}_{\mathcal{F}},$$

where $w_1 = (u_1, x_1)$ and where $\boldsymbol{\Sigma}_{\mathcal{F}} := (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\mathcal{F}})$ is a dynamical system taking values in the combined signal space with

$$\begin{aligned}\mathfrak{B}_{\mathcal{F}} := \{(x_1, u_1, x_2, u_2) \in \mathbb{W}^{\mathbb{T}} | x_1(0) = \mathcal{F}_0(x_2(0)) \text{ and} \\ u_2(t) = \mathcal{F}(x_1(t), u_1(t), x_2(t))\}.\end{aligned}$$

The dynamical system $\boldsymbol{\Sigma}_{c_2}$ is a well-posed controller for $\boldsymbol{\Sigma}_2$ with $\boldsymbol{\Sigma}_2 \times_{w_2} \boldsymbol{\Sigma}_{c_2}$ sharing $w_2 = (u_2, x_2)$. Denote with $\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}}$ the behaviour of the controlled system, then due to the construction of $\boldsymbol{\Sigma}_{\mathcal{F}}$ it follows that $\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}}$ is non-empty and $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$ such that (x_{10}, x_{20}) has a unique continuation in $\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}}$, which is such that $(x_1(t), x_2(t)) \in \mathcal{R}\ \forall t$. Furthermore it holds that $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}}) \subseteq \Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}_{c_1}})$. This holds since $(x_1(t), x_2(t)) \in \mathcal{R}$ holds $\forall t$, which also implies that the output of both systems are the same. Remark that $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}_{c_2}})$ is only a subset of $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}_{c_1}})$, since there could be states x_{10} for which there does not exist $x_{20} \in \mathbb{X}_{20}$. □

Thus based on Proposition 6, we know that we can do exact control refinement (20) for non-singular driving variable systems.

B. Exact control refinement from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}_{DV_a}$

We first derive the static function $\mathcal{S}_a$ mapping transitions of $\boldsymbol{\Sigma}_a$ to the auxiliary input s_a of $\boldsymbol{\Sigma}_{\text{DV}_a}$. From the definition of DV systems, we can also derive the transitions of $\boldsymbol{\Sigma}_{\text{DV}_a}$ indexed with a, which is similar to the derivation of (15).

$$\begin{bmatrix} x_a^+ \\ u_a \end{bmatrix} = M_{a_{\text{right}}}^{-1} A_a x_a + N_a s_a. \tag{23}$$

Multiplying N_a^T on both sides of (23), $\mathcal{S}_a$ is derived as

$$\mathcal{S}_a : s_a = \mathcal{S}_a(x_a^+, u_a, x_a) = N_a^T \begin{bmatrix} x_a^+ \\ u_a \end{bmatrix} - N_a^T M_{a_{\text{right}}}^{-1} A_a x_a. \tag{24}$$

$\mathcal{S}_a$ maps the state evolutions of $\boldsymbol{\Sigma}_a \times_{w_a} \boldsymbol{\Sigma}_{c_a}$ to the auxiliary input s_a for $\boldsymbol{\Sigma}_{\text{DV}_a}$, where $w_a = (u_a, x_a)$. Now, we consider the exact control refinement from the abstract DS to the abstract DV system.

Theorem 7. *Let $\boldsymbol{\Sigma}_a$ be the abstract DS with dynamics (E_a, A_a, B_a, C_a) satisfying the condition of Assumption 1 and let $\boldsymbol{\Sigma}_{DV_a}$ be its related DV system with dynamics $(A_{da}, B_{da}, C_{u_a}, D_{u_a}, C_a)$ such that both systems are initialised with $\mathbb{X}_{a0}$. Then, for any $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$ and based on the static map $\mathcal{S}_a$ (24), there exists a controller $\boldsymbol{\Sigma}_{DV_a}^c \in \mathfrak{C}(\boldsymbol{\Sigma}_{DV_a})$ that is an exact control refinement for $\boldsymbol{\Sigma}_{c_a}$ as defined in Definition 5 with*

$$\Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_a \times \boldsymbol{\Sigma}_{c_a}}\big) = \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{DV_a} \times \boldsymbol{\Sigma}_{DV_a}^c}\big).$$

The proof builds on the construction of a controller $\boldsymbol{\Sigma}_{\text{DV}_a}^c$.

Proof. Denote with x_a and x_a^d the state variables of $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_{\text{DV}_a}$, respectively. Next, we construct the controller $\boldsymbol{\Sigma}_{\text{DV}_a}^c$ that achieves exact control refinement for $\boldsymbol{\Sigma}_{c_a}$ as

$$\boldsymbol{\Sigma}_{\text{DV}_a}^c := (\boldsymbol{\Sigma}_a \times_{w_a} \boldsymbol{\Sigma}_{c_a}) \times_{w_a} \boldsymbol{\Sigma}_{\mathcal{S}_a},$$

where $w_a = (u_a, x_a)$ and where $\boldsymbol{\Sigma}_{\mathcal{S}_a} := (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\mathcal{S}_a})$ is a dynamical system with

$$\begin{aligned}\mathfrak{B}_{\mathcal{S}_a} := \{(x_a, u_a, x_a^d, s_a) \in \mathbb{W}^{\mathbb{T}} | x_a(0) = x_a^d(0) \text{ and} \\ s_a(t) = \mathcal{S}_a(x_a(t+1), u_a(t), x_a(t))\} \text{ as in (24).}\end{aligned}$$

The dynamical system $\boldsymbol{\Sigma}_{\text{DV}_a}^c$ is a well-posed controller for $\boldsymbol{\Sigma}_{\text{DV}_a}$ with $\boldsymbol{\Sigma}_{\text{DV}_a} \times_{w_a^d} \boldsymbol{\Sigma}_{\text{DV}_a}^c$ sharing $w_a^d = (s_a, x_a^d)$. Denote with $\mathfrak{B}_{\boldsymbol{\Sigma}_{\text{DV}_a} \times \boldsymbol{\Sigma}_{\text{DV}_a}^c}$ the behaviour of the controlled system. By construction, we know that the set of the behaviour is non-empty and there is a unique continuation for any $x_{a0}^d \in \mathbb{X}_{a0}$. Further based on the construction of $\boldsymbol{\Sigma}_{\mathcal{S}_a}$, the behaviour is such that $x_a^d(t) = x_a(t), \forall t \in \mathbb{N}$. Additionally, since $\boldsymbol{\Sigma}_a$ and $\boldsymbol{\Sigma}_{\text{DV}_a}$ share the same set of initial states $\mathbb{X}_{a0}$, it holds that $\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma}_a \times \boldsymbol{\Sigma}_{c_a}}\right) = \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma}_{\text{DV}_a} \times \boldsymbol{\Sigma}_{\text{DV}_a}^c}\right)$. □

C. Exact control refinement from $\boldsymbol{\Sigma}_{DV}$ to $\boldsymbol{\Sigma}$

We consider the exact control refinement from $\boldsymbol{\Sigma}_{\text{DV}}$ to $\boldsymbol{\Sigma}$. Suppose we are given a well-posed controller $\boldsymbol{\Sigma}_{\text{DV}}^c$ for $\boldsymbol{\Sigma}_{\text{DV}}$, which shares the free variable s and the state variable x with $\boldsymbol{\Sigma}_{\text{DV}}$. Now we want to design a well-posed controller for $\boldsymbol{\Sigma}$ over $w = (u, x)$, for which we consider the dynamical system $\boldsymbol{\Sigma}_{\mathcal{C}} = (\mathbb{N}, \mathbb{W}, \mathfrak{B})$ over the signal space $\mathbb{W} = \mathbb{U} \times \mathbb{X} \times \mathbb{S}$, the behaviour of which can be defined by

$$\begin{aligned} B_d^T x(t+1) &= B_d^T A_d x(t) + B_d^T B_d s(t) \\ u(t) &= C_u x(t) + D_u s(t). \end{aligned} \tag{25}$$

Then the dynamics of the interconnected system $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_{\mathcal{C}}$ as a function of x and s is derived as

$$\begin{bmatrix} E \\ B_d^T \end{bmatrix} x(t+1) = \begin{bmatrix} A + BC_u \\ B_d^T A_d \end{bmatrix} x(t) + \begin{bmatrix} BD_u \\ B_d^T B_d \end{bmatrix} s(t). \tag{26}$$

Note that $A + BC_u = EA_d$ and $BD_u = EB_d$ by multiplying $M = \begin{bmatrix} E & -B \end{bmatrix}$ on the left-hand side of the two equations in (17). Therefore, (26) is simplified to

$$\begin{bmatrix} E \\ B_d^T \end{bmatrix} x(t+1) = \begin{bmatrix} E \\ B_d^T \end{bmatrix} A_d x(t) + \begin{bmatrix} E \\ B_d^T \end{bmatrix} B_d s(t). \tag{27}$$

Furthermore $\begin{bmatrix}E^T & B_d\end{bmatrix}^T$ has full column rank because the matrix $\begin{bmatrix}M^T & N\end{bmatrix}^T$ is square and has full rank. Recall the requirements for well posedness of a controller as given in Definition 3. Hence $\begin{bmatrix}E^T & B_d\end{bmatrix}^T$ has a left inverse and the dynamics of $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_{\mathcal{C}}$ in (27) can be simplified as

$$x(t+1) = A_d x(t) + B_d s(t),$$

which is exactly the same as the state evolutions of $\boldsymbol{\Sigma}_{\text{DV}}$ as shown in (18). Next we construct $\boldsymbol{\Sigma}_c := \boldsymbol{\Sigma}_{\mathcal{C}} \times_{w^d} \boldsymbol{\Sigma}_{DV}^c$ with $w^d = (s, x^d)$, which is a well-posed controller for $\boldsymbol{\Sigma}$. Then the following theorem regarding the control refinement from $\boldsymbol{\Sigma}_{\text{DV}}$ to $\boldsymbol{\Sigma}$ is developed.

Theorem 8. *Let $\boldsymbol{\Sigma}$ be the concrete DS with dynamics (E, A, B, C) satisfying Assumption 1 and let $\boldsymbol{\Sigma}_{DV}$ be its related DV system with dynamics (A_d, B_d, C_u, D_u, C) such that both systems are initialised with $\mathbb{X}_0$. Then, for any $\boldsymbol{\Sigma}_{DV}^c \in \mathfrak{C}(\boldsymbol{\Sigma}_{DV})$, there exists a controller $\boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma})$ that is an exact control refinement for $\boldsymbol{\Sigma}_{DV}^c$ with*

$$\Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{DV} \times \boldsymbol{\Sigma}_{DV}^c}\big) = \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma} \times \boldsymbol{\Sigma}_c}\right).$$

Proof. Denote with x and x^d the state variables of the $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_{\text{DV}}$, respectively. Next, we construct the controller $\boldsymbol{\Sigma}_c$ that achieves exact control refinement for $\boldsymbol{\Sigma}_{\text{DV}}^c$ as

$$\boldsymbol{\Sigma}_c := \boldsymbol{\Sigma}_{\mathcal{C}} \times_{w^d} \boldsymbol{\Sigma}_{\text{DV}}^c,$$

where $w^d = (s, x^d)$ and the dynamics of $\boldsymbol{\Sigma}_{\mathcal{C}}$ is defined as (25). Then, we can show that the dynamical system $\boldsymbol{\Sigma}_c$ is a well-posed controller for $\boldsymbol{\Sigma}$. Based on the analysis of (27), it is shown that $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_{\mathcal{C}} = \boldsymbol{\Sigma}_{\text{DV}}$ with $w = (u, x)$, then we can derive $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}_{\text{DV}} \times_{w^d} \boldsymbol{\Sigma}_{\text{DV}}^c$. Therefore, we can conclude $\boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma})$ with $\Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma}_{\text{DV}} \times \boldsymbol{\Sigma}_{\text{DV}}^c}\big) = \Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\boldsymbol{\Sigma} \times \boldsymbol{\Sigma}_c}\big)$ follows from $\boldsymbol{\Sigma}_{\text{DV}}^c \in \mathfrak{C}(\boldsymbol{\Sigma}_{\text{DV}})$. □

1) Exact control refinement for descriptor systems: We can now argue that there exists exact control refinement from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}$, as stated in the following result.

Theorem 9. *Consider two DS $\boldsymbol{\Sigma}_a$ (abstract, initialised with $\mathbb{X}_{a0}$) and $\boldsymbol{\Sigma}$ (concrete, initialised with $\mathbb{X}_0$) satisfying Assumption 1 and let $\mathcal{R}$ be a simulation relation from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}$, for which in addition holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$. Then, for any $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$, there exists a controller $\boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma})$ such that*

$$\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma} \times \boldsymbol{\Sigma}_c}\right) \subseteq \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma}_a \times \boldsymbol{\Sigma}_{c_a}}\right).$$

Proof. Based on Assumption 1, we first construct $\boldsymbol{\Sigma}_{\text{DV}}$ and $\boldsymbol{\Sigma}_{\text{DV}_a}$. Then to prove exact control refinement, we need to construct it. This can be done based on the subsequent control refinements given in Theorem 7, Proposition 6 and Theorem 8. □

Theorem 9 claims the existence of such controller $\boldsymbol{\Sigma}_c$ that achieves exact control refinement for $\boldsymbol{\Sigma}_{c_a}$. More precisely, we will show in the proof that the refined controller $\boldsymbol{\Sigma}_c$ is constructive, which provides the solution to Problem 1. Note that assumption 1 on controllability can be relaxed, since the condition $rank(\begin{bmatrix}\sigma E - A & B\end{bmatrix}) = n$ is not required in the proofs.

Example 8. *[Examples 6,7 - continued] Consider the DS of Example 6 and the related DV system in Example 7, and suppose that both systems are initialised within $\mathbb{X}_0 = \{x_0 \mid x_0 \in [-1, 1]^3 \subset \mathbb{R}^3\}$. According to the Silverman-Ho algorithm [9], we can select an abstract DS $\boldsymbol{\Sigma}_a = (E_a, A_a, B_a, C_a)$ that is the minimal realisation of $\boldsymbol{\Sigma}$, is initialised with $\mathbb{X}_{a0} = \mathbb{R}^2$, and where*

$$E_a = \left[\begin{smallmatrix}0 & 0\\ 1 & 0\end{smallmatrix}\right], A_a = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right], B_a = \left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right], C_a = \left[\begin{smallmatrix}0.7\\ 0.2\end{smallmatrix}\right]^T.$$

Similarly, the DV system $\boldsymbol{\Sigma}_{DV_a}$ for $\boldsymbol{\Sigma}_a$ is given as

$$\begin{aligned} x_a(t+1) &= \left[\begin{smallmatrix}0 & 1\\ 0 & 0\end{smallmatrix}\right] x_a(t) + \left[\begin{smallmatrix}0\\ -1\end{smallmatrix}\right] s_a(t), \\ u_a(t) &= \left[\begin{smallmatrix}-1 & 0\end{smallmatrix}\right] x_a(t), y_a(t) = \left[\begin{smallmatrix}0.7 & 0.2\end{smallmatrix}\right] x_a(t). \end{aligned} \tag{28}$$

Further, $\mathcal{R} := \{(x_a, x) \mid x_a = \mathcal{H}x, x_a \in \mathbb{X}_a, x \in \mathbb{X}\}$ is a simulation relation from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}$ with

$$\mathcal{H} = \left[\begin{smallmatrix}0 & 0 & 1\\ 0 & 1 & -1\end{smallmatrix}\right].$$

This can be derived via the two properties in Definition 6. In addition, the following condition holds: $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$. According to Theorem 9, we can refine any $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$ to attain a well-posed controller $\boldsymbol{\Sigma}_c$ for $\boldsymbol{\Sigma}$ that solves Problem 1, as follows: suppose we are given $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$ with dynamics as

$$\left[\begin{smallmatrix}1 & 1\end{smallmatrix}\right] x_a(t+1) = \left[\begin{smallmatrix}0.5 & 0.5\end{smallmatrix}\right] x_a(t) + u_a(t).$$

The controlled system $\boldsymbol{\Sigma}_a \times_{w_a} \boldsymbol{\Sigma}_{c_a}$ is directly derived as

$$\begin{aligned} x_a(t+1) &= \left[\begin{smallmatrix}0 & 1\\ -0.5 & -0.5\end{smallmatrix}\right] x_a(t) \\ y_a(t) &= \left[\begin{smallmatrix}0.7 & 0.2\end{smallmatrix}\right] x_a(t), \end{aligned}$$

with $w_a = (u_a, x_a)$ and $u_a(t) = \left[\begin{smallmatrix}-1 & 0\end{smallmatrix}\right] x_a(t)$. Then $\boldsymbol{\Sigma}_a \times_{w_a} \boldsymbol{\Sigma}_{c_a}$ is stable. According to Theorem 7, we derive the map $\mathcal{S}_a$ for $\boldsymbol{\Sigma}_{DV_a}$ as

$$s_a(t) = \left[\begin{smallmatrix}0 & -1\end{smallmatrix}\right] x_a(t+1) = \left[\begin{smallmatrix}0.5 & 0.5\end{smallmatrix}\right] x_a(t).$$

Next, the interface from $\boldsymbol{\Sigma}_{DV_a}$ to $\boldsymbol{\Sigma}_{DV}$ is developed as $s(t) = s_a(t) - \left[\begin{smallmatrix}0 & 1 & -1\end{smallmatrix}\right] x(t)$. According to Theorem 8, we derive the well-posed controller $\boldsymbol{\Sigma}_c$ as

$$\begin{aligned} \left[\begin{smallmatrix}0 & -1 & 0\end{smallmatrix}\right] x(t+1) &= \left[\begin{smallmatrix}0 & -1 & 1\end{smallmatrix}\right] x(t) + \left[\begin{smallmatrix}0.5 & 0.5\end{smallmatrix}\right] x_a(t) \\ u(t) &= \left[\begin{smallmatrix}0 & 0 & -1\end{smallmatrix}\right] x(t). \end{aligned} \tag{29}$$

Thus, $\boldsymbol{\Sigma}_c \in \mathfrak{C}(\boldsymbol{\Sigma})$ and $\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma} \times \boldsymbol{\Sigma}_c}\right) \subseteq \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\boldsymbol{\Sigma}_a \times \boldsymbol{\Sigma}_{c_a}}\right)$. The interconnected system $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_c$ with $w = (u, x)$, is derived as

$$\begin{aligned} x(t+1) &= \begin{bmatrix}1 & 0 & 1\\ 0 & 1 & -1\\ 0 & 1 & -1\end{bmatrix} x(t) + \begin{bmatrix}0 & 0\\ -0.5 & -0.5\\ 0 & 0\end{bmatrix} x_a(t) \\ y(t) &= \left[\begin{smallmatrix}0 & 0.2 & 0.5\end{smallmatrix}\right] x(t). \end{aligned}$$

Since $(x_a, x) \in \mathcal{R}$, that is $x_a = \mathcal{H}x$, the interconnection $\boldsymbol{\Sigma} \times_w \boldsymbol{\Sigma}_c$ can be simplified by replacing $x_a(t)$ as follows:

$$\begin{aligned} x(t+1) &= \begin{bmatrix}1 & 0 & 1\\ 0 & 0.5 & -1\\ 0 & 1 & -1\end{bmatrix} x(t) \\ y(t) &= \left[\begin{smallmatrix}0 & 0.2 & 0.5\end{smallmatrix}\right] x(t). \end{aligned}$$

Corollary (of Theorem 9)**.** *Consider two DS $\boldsymbol{\Sigma}_a$ (abstract, initialised with $\mathbb{X}_{a0}$) and $\boldsymbol{\Sigma}$ (concrete, initialised with $\mathbb{X}_0$) satisfying Assumption 1 and let $\mathcal{R}$ be a simulation relation from $\boldsymbol{\Sigma}_a$ to $\boldsymbol{\Sigma}$, for which in addition holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}$. Then, for any $\boldsymbol{\Sigma}_{c_a} \in \mathfrak{C}(\boldsymbol{\Sigma}_a)$, there exists a* *controller* $\mathbf{\Sigma}_c$ *such that* $\mathbf{\Sigma} \times \mathbf{\Sigma}_c$ *is* a regular interconnection *and*

$$\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}\right) \subseteq \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}\right).$$

Proof. Based on Theorem 9, we have that there exists a well posed controller $\mathbf{\Sigma}_c' \in \mathfrak{C}(\mathbf{\Sigma})$ such that

$$\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c'}\right) \subseteq \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}\right).$$

Based on Theorem 7, Proposition 6 and Theorem 8 know that $\mathbf{\Sigma}_c'$ is constructed as

$$\mathbf{\Sigma}_c' := \mathbf{\Sigma}_{c_a} \times_{u_a,x_a} \mathbf{\Sigma}_a \times_{u_a,x_a} \mathbf{\Sigma}_{\mathcal{S}_a} \times_{s_a,x_a} \mathbf{\Sigma}_{\mathcal{F}} \times_{s,x} \mathbf{\Sigma}_{\mathcal{C}}$$

Moreover, since $\mathbf{\Sigma}_{c_a}$ is a well-posed controller, it holds that $\mathbf{\Sigma}_{c_a} \times_{u_a,x_a} \mathbf{\Sigma}_a$ is an autonomous system. Furthermore, $\mathbf{\Sigma}_{\mathcal{S}_a}$ is a mapping from x_a, u_a to s_a based on (24). Therefore, it is sufficient to prove that the systems $\mathbf{\Sigma}_{\mathcal{F}} \times_{s,x} \mathbf{\Sigma}_{\mathcal{C}}$ can be replaced by a regular interface denoted as $\mathbf{\Sigma}_{\mathcal{F}_{reg}}$ that yields a regular connection with $\mathbf{\Sigma}$.

Consider the signal space $\mathbb{W} := \mathbb{S}_a \times \mathbb{X}_a \times \mathbb{S} \times \mathbb{X} \times \mathbb{U}$ and let $\mathbf{\Sigma}$ be trivially extended to this signal space, then the interconnection of the systems

$$\mathbf{\Sigma}_{\mathcal{F}} \times_{s,x} \mathbf{\Sigma}_{\mathcal{C}} \times \mathbf{\Sigma}$$

yields a linear dynamical system that defines the behaviour $\mathfrak{B} \subset \mathbb{W}^{\mathbb{T}}$ and for which $\mathfrak{B} \subset \mathfrak{B}_{\mathbf{\Sigma}} \subset \mathbb{W}^{\mathbb{T}}$. Based on Proposition 2 and its corollary there exists a linear dynamical system $\mathbf{\Sigma}_{\mathcal{F}_{reg}}$ for which $\mathbf{\Sigma}_{\mathcal{F}_{reg}} \times \mathbf{\Sigma}$ is a regular interconnection, and for which $\mathfrak{B}_{\mathbf{\Sigma}_{\mathcal{F}_{reg}}\times\mathbf{\Sigma}} = \mathfrak{B}$. Therefore, we have that

$$\mathbf{\Sigma}_c := \mathbf{\Sigma}_{c_a} \times_{u_a,x_a} \mathbf{\Sigma}_a \times_{u_a,x_a} \mathbf{\Sigma}_{\mathcal{S}_a} \times_{s_a,x_a} \mathbf{\Sigma}_{\mathcal{F}_{reg}}$$

is such that $\mathbf{\Sigma}_c \times \mathbf{\Sigma}$ is a regular interconnection and

$$\Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}\right) \subseteq \Pi_{\mathbb{Y}}\left(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}\right).$$

□

Remark 9 (Discussion on constructiveness of the controller synthesis)**.** Given Theorem 9, and its Corollary, we know that a controller refinement exists. We also know that this is constructive as the corresponding proof is constructive. Furthermore, based on [28], we can use this controller to construct a new controller that yields a regular interconnection with the original descriptor system.

While for Theorem 9 controllability is not key, it is needed to achieve a regular controller.

Remark 10 (Comparison to results for standard LTI systems)**.** In comparison to results on control refinement for standard LTI systems the existence of a simulation relation is not sufficient to achieve a well-posed or regular controller. Instead additional controllability conditions are needed that indicated the implementability of the achieved controller.

VI. Approximate control refinement for descriptor systems

In this section we use approximate simulation relations for control refinement. We show that this solves the approximate control refinement as given in Problem 2.

Approximate relationships that do allow for the possibility of error, will certainly provide more freedom in control refinement and in the selection of the abstract model. As a contrast to the exact control refinement, we will focus on Problem 2: approximate control refinement for descriptor systems via the approximate simulation relations in this section. For the solution of Problem 2, we still consider the concrete and abstract DS $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_a$ defined as (1) and (10), respectively. We show that if there exists an approximate simulation relation $\mathcal{R}_\varepsilon$ from $\mathbf{\Sigma}_a$ to $\mathbf{\Sigma}$, and in addition, $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$, then for any $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$, we can always refine $\mathbf{\Sigma}_{c_a}$ to attain a controller $\mathbf{\Sigma}_c$ for $\mathbf{\Sigma}$ such that $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ and $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}))$.

To show this, we can use a lot of the reasoning for exact control refinement. More precisely, we also need the related DV systems $\mathbf{\Sigma}_{\text{DV}}$ and $\mathbf{\Sigma}_{\text{DV}_a}$ as (18) and (19) satisfying $\mathbf{\Sigma}_{\text{DV}} \cong \mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\text{DV}_a} \cong \mathbf{\Sigma}_a$, respectively. For DS $\mathbf{\Sigma}$, $\mathbf{\Sigma}_a$ and their related DV systems $\mathbf{\Sigma}_{\text{DV}}$, $\mathbf{\Sigma}_{\text{DV}_a}$, we combine the following results:

(I) The approximate control refinement for the DV systems:

$$\begin{aligned} &\forall \mathbf{\Sigma}^c_{\text{DV}_a} \in \mathfrak{C}(\mathbf{\Sigma}_{\text{DV}_a}), \exists \mathbf{\Sigma}^c_{\text{DV}} \in \mathfrak{C}(\mathbf{\Sigma}_{\text{DV}}), \text{ s.t.} \\ &\Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\mathbf{\Sigma}_{\text{DV}}\times\mathbf{\Sigma}^c_{\text{DV}}}\big) \subseteq \mathcal{E}_\varepsilon\big(\Pi_{\mathbb{Y}}\big(\mathfrak{B}_{\mathbf{\Sigma}_{\text{DV}_a}\times\mathbf{\Sigma}^c_{\text{DV}_a}}\big)\big); \end{aligned} \tag{30}$$

(II) The exact control refinement from $\mathbf{\Sigma}_a$ to $\mathbf{\Sigma}_{\text{DV}_a}$ as given in Equation (21);

(III) The exact control refinement from $\mathbf{\Sigma}_{\text{DV}}$ to $\mathbf{\Sigma}$ as given in Equation (22).

When compared to the elements presented for the exact case, only the first condition is new and needs to be proven. Again, it will be shown that the combination of the elements (I)-(III) also implies the construction of the approximate control refinement for the concrete and abstract DS.

Let us prove element (I) next.

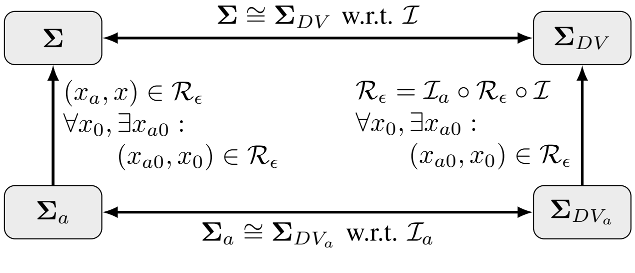

$\mathbf{\Sigma} \cong \mathbf{\Sigma}_{DV}$ w.r.t. $\mathcal{I}$
$\mathbf{\Sigma}$
$\mathbf{\Sigma}_{DV}$
$(x_a, x) \in \mathcal{R}_\epsilon$
$\forall x_0, \exists x_{a0}$:
$(x_{a0}, x_0) \in \mathcal{R}_\epsilon$
$\mathcal{R}_\epsilon = \mathcal{I}_a \circ \mathcal{R}_\epsilon \circ \mathcal{I}$
$\forall x_0, \exists x_{a0}$:
$(x_{a0}, x_0) \in \mathcal{R}_\epsilon$
$\mathbf{\Sigma}_a$
$\mathbf{\Sigma}_{DV_a}$
$\mathbf{\Sigma}_a \cong \mathbf{\Sigma}_{DV_a}$ w.r.t. $\mathcal{I}_a$

Fig. 6: Connection between DS and DV systems for approximate control refinement. The schematic connection for approximate control refinement is very similar to that of exact control refinement given in Fig. 4.

A. Approximate control refinement for the DV systems

Similar to the exact control refinement case, we can again infer the relation between the DV systems next, as depicted in Fig. 6. We can thus derive that $\mathcal{R}_\varepsilon$ is an approximate simulation relation from $\mathbf{\Sigma}_{\mathrm{DV}_a}$ to $\mathbf{\Sigma}_{\mathrm{DV}}$.

Proposition 10. *Let $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ be two non-singular DS defined over the same output space $\mathbb{Y}$ with dynamics (I_1, A_1, B_1, C_1) and (I_2, A_2, B_2, C_2), which are initialised with $\mathbb{X}_{10}$ and $\mathbb{X}_{20}$, respectively. If there exists an approximate relation $\mathcal{R}_\varepsilon \subseteq \mathbb{X}_1 \times \mathbb{X}_2$ such that*

1) *$\mathcal{R}_\varepsilon$ is an approximate simulation relation from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$ as in Def. 7, and*
2) *$\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$ s.t. $(x_{10}, x_{20}) \in \mathcal{R}_\varepsilon$,*

then for any controller $\mathbf{\Sigma}_{c_1} \in \mathfrak{C}(\mathbf{\Sigma}_1)$, there exists a controller $\mathbf{\Sigma}_{c_2} \in \mathfrak{C}(\mathbf{\Sigma}_2)$ that is an approximate control refinement for $\mathbf{\Sigma}_{c_1}$, as defined in Def. 5, i.e.,

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2 \times \mathbf{\Sigma}_{c_2}}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1 \times \mathbf{\Sigma}_{c_1}})).$$

Proof. Consider an approximate simulation relation $\mathcal{R}_\varepsilon$ from $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_2$. Then there also exists an interface function $\mathcal{F} : \mathbb{U}_1 \times \mathbb{X}_1 \times \mathbb{X}_2 \to \mathbb{U}_2$ related to $\mathcal{R}_\varepsilon$ that satisfies condition (1) of Def. 8. In addition, due to the initial states in condition (2), there exists a map $\mathcal{F}_0^\varepsilon : \mathbb{X}_{20} \to \mathbb{X}_{10}$ such that for all $x_{20} \in \mathbb{X}_{20}$ it holds that $(\mathcal{F}_0^\varepsilon(x_{20}), x_{20}) \in \mathcal{R}_\varepsilon$.
Next, we construct the controller $\mathbf{\Sigma}_{c_2}$ that achieves approximate control refinement for $\mathbf{\Sigma}_{c_1}$ as

$$\mathbf{\Sigma}_{c_2} := (\mathbf{\Sigma}_1 \times_{w_1} \mathbf{\Sigma}_{c_1}) \times_{w_1} \mathbf{\Sigma}_{\mathcal{F}}^\varepsilon,$$

where $w_1 = (u_1, x_1)$ and where $\mathbf{\Sigma}_{\mathcal{F}}^\varepsilon := (\mathbb{N}, \mathbb{W}, \mathfrak{B}_{\mathcal{F}}^\varepsilon)$ is a dynamical system taking values in the combined signal space with

$$\mathfrak{B}_{\mathcal{F}}^\varepsilon := \{(x_1, u_1, x_2, u_2) \in \mathbb{W}^{\mathbb{T}} | x_1(0) = \mathcal{F}_0^\varepsilon(x_2(0)) \text{ and } u_2(t) = \mathcal{F}(x_1(t), u_1(t), x_2(t))\}.$$

Similarly, the dynamical system $\mathbf{\Sigma}_{c_2}$ is a well-posed controller for $\mathbf{\Sigma}_2$ that achieves approximate control refinement. Due to the construction of $\mathbf{\Sigma}_{\mathcal{F}}^\varepsilon$ it follows that $\mathfrak{B}_{\mathbf{\Sigma}_2 \times \mathbf{\Sigma}_{c_2}}$ is non-empty and $\forall x_{20} \in \mathbb{X}_{20}, \exists x_{10} \in \mathbb{X}_{10}$ such that (x_{10}, x_{20}) has a unique continuation in $\mathfrak{B}_{\mathbf{\Sigma}_2 \times \mathbf{\Sigma}_{c_2}}$ for which it holds that $(x_1(t), x_2(t)) \in \mathcal{R}_\epsilon$, $\forall t$. Furthermore it holds that $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_2 \times \mathbf{\Sigma}_{c_2}}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_1 \times \mathbf{\Sigma}_{c_1}}))$, because for all x_{20} there exists continuations in both $\mathfrak{B}_{\mathbf{\Sigma}_2 \times \mathbf{\Sigma}_{c_2}}$ and $\mathfrak{B}_{\mathbf{\Sigma}_1 \times \mathbf{\Sigma}_{c_1}}$ such that the output difference is bounded as $d(y_1, y_2) \leq \epsilon$ over time. □

Since the DV systems $\mathbf{\Sigma}_{\mathrm{DV}}$ and $\mathbf{\Sigma}_{\mathrm{DV}_a}$ share the same initial states as the respective DS $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_a$, it also holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$. According to Proposition 10, we know that we can do approximate control refinement, as shown in (30).

B. Approximate control refinement for descriptor systems

We can now again combine the different theorems to prove the control refinement in the approximate setting. Based on Proposition 10 for condition (I) of the approximate control refinement, Theorem 7 for condition (II) and based on Theorem 8 for condition (III), we develop the following theorem as a solution for Problem 2.

Theorem 11. *Consider two DS $\mathbf{\Sigma}_a$ (abstract, initialised with $\mathbb{X}_{a0}$) and $\mathbf{\Sigma}$ (concrete, initialised with $\mathbb{X}_0$) satisfying Assumption 1 and let $\mathcal{R}_\varepsilon$ be an approximate simulation relation from $\mathbf{\Sigma}_a$ to $\mathbf{\Sigma}$, for which in addition holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$. Then, for any $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$, there exists a controller $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ such that*

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma} \times \mathbf{\Sigma}_c}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a \times \mathbf{\Sigma}_{c_a}})).$$

Proof. Based on Assumption 1, we first construct $\mathbf{\Sigma}_{\mathrm{DV}}$ and $\mathbf{\Sigma}_{\mathrm{DV}_a}$. Then to prove this we need to construct the approximate control refinement. Similar to Theorem 9, this can also be done based on the subsequent control refinements given in Theorem 7, Proposition 10 and Theorem 8. □

Theorem 11 claims the existence of such controller $\mathbf{\Sigma}_c$ that achieves approximate control refinement for $\mathbf{\Sigma}_{c_a}$. As before, since each of the theorems and propositions are constructive for the control refinement, it follows that the combined controller can be constructed.

Corollary (of Theorem 11)**.** *Consider two DS $\mathbf{\Sigma}_a$ (abstract, initialised with $\mathbb{X}_{a0}$) and $\mathbf{\Sigma}$ (concrete, initialised with $\mathbb{X}_0$) satisfying Assumption 1 and let $\mathcal{R}_\varepsilon$ be an approximate simulation relation from $\mathbf{\Sigma}_a$ to $\mathbf{\Sigma}$, for which in addition holds that $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$. Then, for any $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$, there exists a controller $\mathbf{\Sigma}_c$ such that $\mathbf{\Sigma} \times \mathbf{\Sigma}_c$ is* a regular interconnection *and*

$$\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma} \times \mathbf{\Sigma}_c}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a \times \mathbf{\Sigma}_{c_a}})).$$

The proof follows the same reasoning as the proof of the corollary of Theorem 9. Again, similarly to Remark 9, we can also construct this controller based on [28].

VII. Computational aspects and case study

In this section, we show how simulation functions and Lyapunov-like functions can be used to find an approximate simulation relation and to construct a refined controller. In order to clarify the discussion we also consider an illustrative example.

A. An interface for driving variable systems

Let two driving variable systems be given, the concrete (original) $\mathbf{\Sigma}_{\mathrm{DV}}$ and an abstraction $\mathbf{\Sigma}_{\mathrm{DV}_a}$. Recall that these driving variable systems are essentially standard difference equations, for which there exist methods to compute interface functions [11], [12].

Consider next the definition of simulation function [11], [12]: these functions will define corresponding approximate simulation relations between two systems. A simulation function is a positive function that bounds the distance between output behaviours and that is non-increasing under the parallel evolution of the two systems.

Definition 9 (Simulation function [14])**.** Let $\mathbf{\Sigma}_a$ and $\mathbf{\Sigma}$ be two non-singular DS defined over the same output space $\mathbb{Y}$ with dynamics (I_a, A_a, B_a, C_a) and (I, A, B, C). A function $\mathcal{S} : \mathbb{X}_a \times \mathbb{X} \to \mathbb{R}^+ \cup \{+\infty\}$ is called a simulation function of

$\mathbf{\Sigma}_a$ by $\mathbf{\Sigma}$ if its sub-level sets are closed, and for all $(x_a, x) \in \mathbb{X}_a \times \mathbb{X}$:

$$\mathcal{S}(x_a, x) \geq \max\left\{ d(C_1 x_a, C_2 x), \sup_{x_a^+} \inf_{x^+} \mathcal{S}(x_a^+, x^+) \right\},$$

where $d(\cdot)$ denotes the Euclidean distance.

We recall the following proposition [12, Theorem 22].

Proposition 12. *Let $\mathcal{S}$ be a simulation function of $\mathbf{\Sigma}_a$ by $\mathbf{\Sigma}$, then, for all $\varepsilon \geq 0$,*

$$\mathcal{R}_\varepsilon = \{(x_a, x) \in \mathbb{X}_a \times \mathbb{X} \mid \mathcal{S}(x_a, x) \leq \varepsilon\} \tag{31}$$

is an approximate simulation relation of $\mathbf{\Sigma}_a$ by $\mathbf{\Sigma}$ of precision ε.

Suppose that a potential interface function is given by

$$\mathcal{F}(u_a, x_a, x) = Ru_a + Qx_a + K(x - Px_a), \tag{32}$$

such that there exists a positive semi-definite matrix M

$$M \succeq C^T C, \; (A + BK)^T M (A + BK) \preceq \lambda^2 M \tag{33}$$

with $\lambda \in (0, 1)$. The right inequality implies that K is a stabilising state feedback for the original system $\mathbf{\Sigma}$. Furthermore, matrices Q and P are such that

$$PA_a = AP + BQ, \; C_a = CP. \tag{34}$$

This constrained Sylvester equations (34) can be solved via either the Kronecker product [21] or an RQ factorisation, respectively.

We are now ready to quantify a simulation relation for these discrete-time systems similar to [11] for continuous-time systems, based on the notion of Lyapunov functions for discrete-time systems.

Proposition 13. *The function $\mathcal{F}(u_a, x_a, x)$ in* (32) *subject to* (33) *and* (34) *is an interface function for the ϵ-approximate simulation relation $\mathcal{R}_\epsilon$, computed as in Equation* (31) *from the simulation function*

$$\mathcal{S}(x_a, x) = \max(\mathcal{V}(x_a, x), \gamma(u_{\max})), \tag{35}$$

with Lyapunov-like auxiliary function

$$\mathcal{V}(x_a, x) = \sqrt{(x - Px_a)^T M (x - Px_a)} \tag{36}$$

and with a γ function defined as

$$\gamma(r) = r \left\| \sqrt{M}(BR - PB_a) \right\|_2 / (1 - \lambda),$$

for $u_{\max} = \max_{u_a \in \mathbb{U}_a} \|u_a\|$.

The proof of this proposition can be found in the Appendix. The construction here is different to the hierarchical control given in [11], [13], which specifically deals with continuous-time and non-singular systems, and with the notion of exact bisimulation. The function $\mathcal{V}$ is used as a Lyapunov-like auxiliary function, much like those used for the theory of input-to-state stability [15], [17] for discrete-time systems.

Next, we employ model reduction methods to attain the abstract system first, and then solve the matrix equation in (34) to derive the projection matrix P, so as to establish connections between concrete and abstract systems.

Consider the concrete DS $\mathbf{\Sigma}$ with dynamics (E, A, B, C) defined as

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.5 \end{bmatrix}^T.$$

The corresponding concrete DV system $\mathbf{\Sigma}_{\text{DV}}$ with dynamics characterised by (A_d, B_d, C_u, D_u, C), is

$$A_d = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix}, B_d = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, C_u = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}^T, D_u = 0.$$

Selecting a stabilizing $K = \begin{bmatrix} 0.1262 & -0.8327 & 0.9843 \end{bmatrix}$, this results in a stable matrix $A_d + B_d K$. Afterwards, based on a simple model order reduction step, we obtain the abstract DS $\mathbf{\Sigma}_a = (E_a, A_a, B_a, C_a)$ and its related abstract DV system $\mathbf{\Sigma}_{\text{DV}_a} = (A_{da}, B_{da}, C_{u_a}, D_{u_a}, C_a)$, where

$$E_a = \begin{bmatrix} -0.705 & 0.709 \\ 0 & 0 \end{bmatrix}, A_a = \begin{bmatrix} -0.051 & -0.29 \\ -1.429 & 1.499 \end{bmatrix}, B_a = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

$$C_a = \begin{bmatrix} 0.889 \\ -0.747 \end{bmatrix}^T, A_{da} = \begin{bmatrix} -0.051 & 0.123 \\ -0.123 & -0.287 \end{bmatrix}, B_{da} = \begin{bmatrix} -1.683 \\ -1.675 \end{bmatrix}.$$

$$C_{ua} = \begin{bmatrix} -1.429 \\ 1.499 \end{bmatrix}^T, D_{ua} = 0.$$

According to Proposition 13, we can design a Lyapunov-like auxiliary function $\mathcal{V}(x_a, x)$ together with the simulation function $\mathcal{S}(x_a, x)$. Afterwards, based on the transitivity of relations and initialisation conditions, the approximate simulation relation from $\mathbf{\Sigma}_a$ to $\mathbf{\Sigma}$ is immediately defined as

$$\mathcal{R}_\varepsilon = \{(x_a, x) \in \mathbb{X}_a \times \mathbb{X} \mid \mathcal{S}(x_a, x) \leq \varepsilon\},$$

with $\varepsilon = \mathcal{S}(x_{a0}, x_0)$, and where in addition $\forall x_0 \in \mathbb{X}_0, \exists x_{a0} \in \mathbb{X}_{a0}$ s.t. $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$. The related interface in (32) for the DV framework depends on matrices P, Q and R, whose values are sought to obtain a small simulation function and hence a large approximate simulation relation, as in Propositions 13 and 12. This gives

$$P = \begin{bmatrix} -1.1597 & 2.4387 \\ 1.5254 & -0.9658 \\ 1.4005 & -1.5960 \end{bmatrix}, Q = \begin{bmatrix} -0.0410 \\ -0.4645 \end{bmatrix}^T, R = 0.955 .$$

Now, let us consider a controller $\mathbf{\Sigma}_{c_a} \in \mathfrak{C}(\mathbf{\Sigma}_a)$ defined as

$$\begin{bmatrix} 1 & 1 \end{bmatrix} x_a(t+1) = \begin{bmatrix} 1 & 1 \end{bmatrix} x_a(t) + u_a(t),$$

for which the interconnected system $\mathbf{\Sigma}_a \times \mathbf{\Sigma}_{c_a}$ is

$$x_a(t+1) = \begin{bmatrix} -0.179 & 1.458 \\ -0.25 & 1.04 \end{bmatrix} x_a(t);$$
$$y_a(t) = \begin{bmatrix} 0.1 & 0.2 & 0.5 \end{bmatrix} x_a(t),$$

with $u_a(t) = C_{ua} x_a(t) = \begin{bmatrix} -1.429 & 1.499 \end{bmatrix} x_a(t)$. Note that $\Sigma_a \times \Sigma_{c_a}$ is stable. Then according to Theorem 7, we derive the map $\mathcal{S}_a$ for $\mathbf{\Sigma}_{\text{DV}_a}$ as

$$s_a(t) = \mathcal{S}_a(x_a(t+1), u_a(t), x_a(t)) = \begin{bmatrix} 0.076 & -0.793 \end{bmatrix} x_a(t).$$

The controlled abstract DV system is the same as $\mathbf{\Sigma}_a \times \mathbf{\Sigma}_{c_a}$. Finally, according to Theorem 11, we derive the refined well-posed controller $\mathbf{\Sigma}_c$ for $\mathbf{\Sigma}$ as

$$\begin{bmatrix} 0 & -1 & 0 \end{bmatrix} x(t+1) = Kx(t) + \begin{bmatrix} 0.07 & -0.763 \end{bmatrix} x_a(t);$$
$$u(t) = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} x(t).$$

The controller $\mathbf{\Sigma}_c$ is a regular controller for $\mathbf{\Sigma}$, this can be check be verifying that the combined systems has full row rank [28].

Remark 11. The abstract system can be computed either via model order reduction techniques for descriptor systems [5] or by applying well-known model order techniques for the equivalent driving variable system such as balanced truncation, proper orthogonal decomposition (POD) and Hankel norm model reduction. Further research is needed to investigate tailored, optimal model reduction methods to compute these abstractions.

For the ensuing experiments, we select initial states $x_{a0} = \begin{bmatrix} 0.3 & 0.3 \end{bmatrix}^T$ and $x_0 = \begin{bmatrix} 0.4 & 0.2 & -0.04 \end{bmatrix}^T$ such that $(x_{a0}, x_0) \in \mathcal{R}_\varepsilon$, with $\varepsilon = 0.0667$. The simulations of the closed loop system are shown in Fig. 7. Since the two controlled systems converge fast, we only show closed-loop trajectories until $t = 15$. As we can see from Fig. 7, the distance between the two controlled DS is within the error bound

$$\varepsilon = \max(\mathcal{V}(x_{a0}, x_0), \gamma(s_{a\max})) = 0.0667.$$

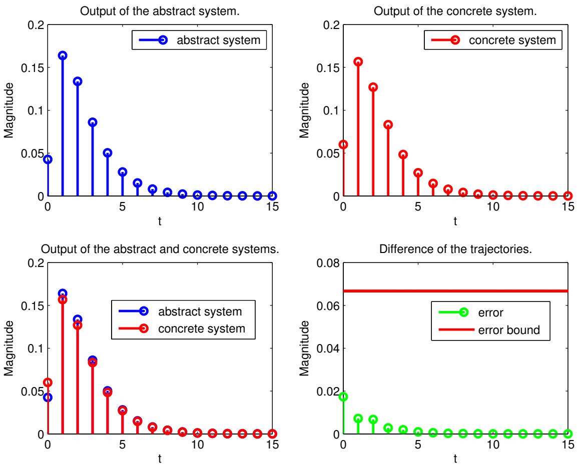


Fig. 7: Closed loop simulation results.

On the other hand, we consider the open loop simulation result by choosing a random signal s_a to Σ_{DV_a} satisfying $s_{a\max} \leq 0.3$. The simulation result is shown in Fig. 8 with $\varepsilon = 0.093$. It can be seen from Fig. 8 that the distance between the output behaviour of the abstract and concrete DS is bounded within $\varepsilon = 0.093$. Finally, we can conclude that $\mathbf{\Sigma}_c \in \mathfrak{C}(\mathbf{\Sigma})$ and $\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}\times\mathbf{\Sigma}_c}) \subseteq \mathcal{E}_\varepsilon(\Pi_{\mathbb{Y}}(\mathfrak{B}_{\mathbf{\Sigma}_a\times\mathbf{\Sigma}_{c_a}}))$.

VIII. Conclusions and Future Work

In this paper, we have dealt with the formal control design problems over complex descriptor systems by developing and analysing a control refinement approach.

The behavioural approach has been used to formulate the notion of well-posed regular controllers for descriptor systems. In order to design controllers on simpler systems, we have leveraged the notion of control refinement. Both exact and approximate simulation relations have been used for, respectively, exact and approximate control refinement over descriptor systems. We have proven that for any well-posed controller of the abstract descriptor system, it can be refined to a well-posed controller for the concrete descriptor system. Furthermore, requiring controllability of the system, we can also show that a regular controller can be constructed.

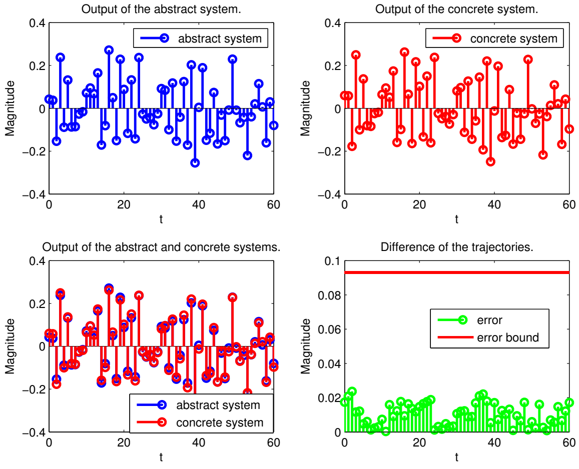


Fig. 8: Open loop simulation results.

Future research will target extensions to nonlinear descriptor systems, and connections to classical results in perturbation theory. For nonlinear systems and control methods further research is needed on compatibility conditions of controllers.

Proof of Proposition 13

In the following, let us detail the notions of simulation function and interface for discrete-time non-singular system based on a Lyapunov-like auxiliary function and a level set.

First we iterate some general conditions from input-to-state stability, to which $\mathbf{\Sigma}_a$, $\mathbf{\Sigma}$ and $\mathcal{V}$ in (36) conform, and use it to prove some properties. Let two systems be given

$$\mathbf{\Sigma}_1 : z^+ = h(z, v), \qquad y_1 = k(z) \tag{37}$$

$$\mathbf{\Sigma}_2 : x^+ = f(x, u), \qquad y_2 = g(x) \tag{38}$$

with the shared output space $\mathbb{Y}$ and states x, z and with the respective inputs u, v. Let a Lyapunov-like auxiliary function $\mathcal{V} : \mathbb{Z}\times\mathbb{X} \to \mathbb{R}^+$ together with a function $u_{\mathcal{V}} : \mathbb{V}\times\mathbb{Z}\times\mathbb{X} \mapsto \mathbb{U}$ be given such that for all $(z, x) \in \mathbb{Z} \times \mathbb{X}$,

$$\mathcal{V}(z, x) \geq \|k(z) - g(x)\| \tag{39}$$

and for all $(v, z, x) \in \mathbb{V} \times \mathbb{Z} \times \mathbb{X}$,

$$\begin{aligned} \mathcal{V}(h(z, v), f(x, \mathcal{F}(v, z,x))) - \mathcal{V}(z, x) \leq \\ - \alpha(\mathcal{V}(z, x)) + \sigma(\|v\|). \end{aligned} \tag{40}$$

In (40), α is a $\mathcal{K}_\infty$ function and σ is a $\mathcal{K}$ function. These functions belong to the special class of comparison functions, known as class $\mathcal{K}$ functions [17], and as defined next.

Definition 10. *A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$.*

One property of $\mathcal{K}_\infty$ function that will be used later is $\alpha \in \mathcal{K}_\infty \Rightarrow \alpha^{-1} \in \mathcal{K}_\infty$, where α^{-1} denotes the inverse function of α.

Lemma 14. *[15] For any $\mathcal{K}_\infty$ function α there is a $\mathcal{K}_\infty$ function $\hat{\alpha}$ satisfying*
1. $\hat{\alpha}(s) \leq \alpha(s), \forall s \geq 0$;
2. $\eta - \hat{\alpha} \in \mathcal{K}$
where η denotes the identity function or identity map, i.e., $\eta(x) = x$.

Proposition 15. *Let $\mathcal{V}$ be a Lyapunov-like auxiliary function and $u_{\mathcal{V}}$ be a function such that (39) and (40) hold. Then, for bounded inputs $\|v\| \leq v_{\max}$, it follows that*

$$\mathcal{S}(z,x) = \max\{\mathcal{V}(z,x), \gamma(v_{\max})\} \tag{41}$$

is a simulation function of Σ_1 by Σ_2, and $\mathcal{F}$ is an interface from Σ_1 to Σ_2. The constructed γ function is given as

$$\gamma(r) = \hat{\alpha}^{-1}\left(\frac{\sigma(r)}{c}\right)$$

with $c \in (0,1]$. $\hat{\alpha}$ is the $\mathcal{K}_\infty$ function chosen according to Lemma 14.

Proof. (of Proposition 15): Consider the Lyapunov-like auxiliary function $\mathcal{V}(z,x)$ satisfying (39) and (40). First, we denote $\mathcal{V}(h(z,v), f(x, \mathcal{F}(v,z,x)))$ by $\mathcal{V}^+(z,x)$ for convenience. Since $\hat{\alpha}$ is the $\mathcal{K}_\infty$ function chosen as Lemma 14, we have $\hat{\alpha}(s) \leq \alpha(s), \forall s \geq 0$. Therefore,

$$\mathcal{V}^+(z,x) - \mathcal{V}(z,x) \leq -\hat{\alpha}(\mathcal{V}(z,x)) + \sigma(\|v\|). \tag{42}$$

For any input sequence v, consider the level set

$$D = ((z,x)|\mathcal{V}(z,x) \leq b)$$

where $b = \hat{\alpha}^{-1}\left(\frac{\sigma(v_{\max})}{c}\right) = \gamma(v_{\max})$. First we prove that when $(z(t_0), x(t_0)) \in D$, $(z(t), x(t)) \in D, \forall t \geq t_0$.

Assume that $(z(t_0), x(t_0)) \in D$, $\mathcal{V}(z(t_0), x(t_0)) \leq b$. With the inequality $\sigma(\|v\|) \leq \sigma(v_{\max})$, we transform (42) into the following form:

$$\begin{aligned}\mathcal{V}^+(z(t_0), x(t_0)) \leq -(1-c)\hat{\alpha}(\mathcal{V}(z(t_0), x(t_0)))+ \\ \tilde{\alpha}(\mathcal{V}(z(t_0), x(t_0))) + \sigma(v_{\max})\end{aligned}$$

where $\tilde{\alpha} = \eta - c\hat{\alpha}$. Since $\eta - \hat{\alpha} \in \mathcal{K}$ as Lemma 14. In addition $\hat{\alpha} \in \mathcal{K}$, we have $(1-c)\hat{\alpha} \in \mathcal{K}$. Therefore, we can conclude that $\tilde{\alpha} = \eta - c\hat{\alpha} = \eta - \hat{\alpha} + (1-c)\hat{\alpha} \in \mathcal{K}$.

Since $c\hat{\alpha}(\mathcal{V}(z(t_0), x(t_0))) \leq c\hat{\alpha}(b) = \sigma(v_{\max})$ and $c\tilde{\alpha}(\mathcal{V}(z(t_0), x(t_0))) \leq c\tilde{\alpha}(b)$, we have

$$\begin{aligned}\tilde{\alpha}(\mathcal{V}(z(t_0), x(t_0))) + \sigma(v_{\max}) \leq& \tilde{\alpha}(b) + \sigma(v_{\max}) = \\ & b - c\hat{\alpha}(b) + \sigma(v_{\max}) = b.\end{aligned}$$

Therefore,

$$\mathcal{V}^+(z(t_0), x(t_0)) \leq -(1-c)\hat{\alpha}(\mathcal{V}(z(t_0), x(t_0))) + b \leq b.$$

By induction, we can show that $(z(t_0+j), x(t_0+j)) \in D, \forall j \in \mathbb{N}$, that is,$(z(t), x(t)) \in D, \forall t \geq t_0$.
Now let $t_0 = \min\{t \in \mathbb{N}_0 | (z(t), x(t)) \in D\} < \infty$. Then

$$\mathcal{V}(z(t), x(t)) \leq \gamma(v_{\max}), \forall t \geq t_0.$$

For $0 \leq t < t_0$, we have $c\hat{\alpha}(\mathcal{V}(z(t), x(t))) > c\hat{\alpha}(b) = \sigma(v_{\max})$. Therefore, $\forall 0 \leq t < t_0$, we have

$$\mathcal{V}^+(z(t), x(t)) - \mathcal{V}(z(t), x(t)) \leq -(1-c)\hat{\alpha}(\mathcal{V}(z(t), x(t))) \leq 0.$$

We have proven that if $(z(0), x(0)) \in D$, it will always remain in the level set and $(z(t), x(t)) \in D, \forall t \in \mathbb{N}$. And if $(z(0), x(0)) \notin D$, $\mathcal{V}(z(t), x(t))$ will decrease until $(z(t), x(t))$ gets in the level set and remains there.

Thus, by truncating the Lyapunov-like auxiliary function $\mathcal{V}(z,x)$ by the level set $\gamma(v_{\max})$, we construct the simulation function (cf. Definition 9) as

$$\mathcal{S}(z,x) = \max(\mathcal{V}(z,x), \gamma(v_{\max})).$$

□

We can now leverage Proposition 15, to prove that Proposition 13 holds. According to equation (33) and (34), we have

$$\mathcal{V}(x_a, x) \geq \sqrt{(x - Px_a)^T C^T C(x - Px_a)} = \|Cx - C_a x_a\|.$$

Thus, inequality (39) holds. To prove that inequality (40) holds, we have

$$\begin{aligned}& x^+ - Px_a^+ = Ax + B[Ru_a + Qx_a + K(x - Px_a)] \\ & - P(A_a x_a + B_a u_a) \\ & = (A + BK)(x - Px_a) + (BR - PB_a)u_a\end{aligned}$$

where x^+ and z_a^+ denote the next states of x and x_a respectively. Therefore,

$$\begin{aligned}& \mathcal{V}^+(x_a, x) - \mathcal{V}(x_a, x) = \\ & \sqrt{(x^+ - Px_a^+)^T M(x^+ - Px_a^+)} \\ & - \sqrt{(x - Px_a)^T M(x - Px_a)} \\ & = \left\|\sqrt{M}[(A + BK)(x - Px_a) + (BR - PB_a)u_a]\right\| - \\ & \sqrt{(x - Px_a)^T M(x - Px_a)}.\end{aligned}$$

From the triangle inequality of norms, we know that

$$\begin{aligned}& \mathcal{V}^+(x_a, x) - \mathcal{V}(x_a, x) \\ & \leq \sqrt{[(A + BK)(x - Px_a)]^T M[(A + BK)(x - Px_a)]} \\ & - \sqrt{(x - Pz)^T M(x - Px_a)} + \left\|\sqrt{M}(BR - PB_a)u_a\right\| \\ & \leq (\lambda - 1)\mathcal{V}(x_a, x) + \left\|\sqrt{M}(BR - PB_a)u_a\right\|.\end{aligned}$$

The second inequality follows from inequality (33).
Hence, $\mathcal{V}(x_a, x)$ satisfies the two conditions (39) and (40) and thus it is a Lyapunov-like auxiliary function. Consequently, according to Proposition 15, γ is the $\mathcal{K}$ function defined as

$$\gamma(r) = \frac{\left\|\sqrt{M}(BR - PB_a)\right\|_2}{1 - \lambda} r. \tag{43}$$

and the function

$$\mathcal{S}(x_a, x) = \max(\mathcal{V}(x_a, x), \gamma(u_{\max}))$$

is a simulation function for the associated interface (32).

References

[1] A. C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.

[2] A. Arnold and J. Plaice. *Finite transition systems: semantics of communicating systems*. Prentice Hall International (UK) Ltd., 1994.

[3] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[4] C. Belta, B. Yordanov, and E. A. Gol. *Formal Methods for Discrete-Time Dynamical Systems*, volume 89. Springer, 2017.

[5] X. Cao. Hankel norm model reduction for discrete-time descriptor systems. Master's thesis, Eindhoven University of Technology.

[6] X. Cao, M. Saltik, and S. Weiland. Hankel model reduction for descriptor systems. In *2015 54th IEEE CDC*, pages 4668–4673, 2015.

[7] F. Chen, S. Haesaert, A. Abate, and S. Weiland. Control refinement for discrete-time descriptor systems: a behavioural approach via simulation relations. *IFAC 20th IFAC World Congress*, 50(1):15822–15827, 2017.

[8] E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT press, 1999.

[9] L. Dai. *Singular control systems*. Springer-Verlag New York, Inc., 1989.

[10] G. E. Fainekos, A. Girard, and G. J. Pappas. Hierarchical synthesis of hybrid controllers from temporal logic specifications. In *International Workshop on HSCC*, pages 203–216, 2007.

[11] A. Girard and G. J. Pappas. Approximate bisimulations for constrained linear systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 4700–4705. IEEE, 2005.

[12] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *Automatic Control, IEEE Transactions on*, 52(5):782–798, 2007.

[13] A. Girard and G. J. Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566–571, 2009.

[14] A. Girard and G. J. Pappas. Approximate bisimulation: A bridge between computer science and control theory. *European Journal of Control*, 17(5):568–578, 2011.

[15] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[16] A. A. Julius and A. J. v. d. Schaft. Compatibility of behavior interconnections. In *2003 European Control Conference (ECC)*, pages 2607–2612, Sept. 2003.

[17] H. K. Khalil and J. Grizzle. *Nonlinear systems*. Prentice hall New Jersey, 1996.

[18] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.

[19] P. Kunkel and V. L. Mehrmann. *Differential-algebraic equations: analysis and numerical solution*. European Mathematical Society, 2006.

[20] O. Kupferman and M. Y. Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314, 2001.

[21] A. J. Laub. *Matrix analysis for scientists and engineers*. SIAM, 2005.

[22] N. Y. Megawati and A. v. d. Schaft. Bisimulation equivalence of differential-algebraic systems. *International Journal of Control*, pages 1–11, 2017.

[23] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[24] S. Weiland. *Theory of Approximation and Disturbnace Attenuation for Linear Systems*. University of Groningen, 1991.

[25] J. Willems and H. Trentelman. On quadratic differential forms. *SIAM Journal on Control and Optimization*, 36(5):1703–1749, 1998.

[26] J. C. Willems. Models for Dynamics. In U. Kirchgraber and H. O. Walther, editors, *Dynamics Reported: A Series in Dynamical Systems and Their Applications*, Dynamics Reported, pages 171–269. Vieweg+Teubner Verlag, Wiesbaden, 1989.

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

[28] J. C. Willems. On interconnections, control, and feedback. *IEEE Transactions on Automatic control*, 42(3):326–339, 1997.

[29] J. C. Willems. The behavioral approach to open and interconnected systems. *Control Systems, IEEE*, 27(6):46–99, 2007.

[30] J. C. Willems and J. W. Polderman. *Introduction to mathematical systems theory: a behavioral approach*. Springer Science & Business Media, 2013.

Sofie Haesaert is an Assistant Professor at the Control Systems group, Department of Electrical Engineering, Eindhoven University of Technology. From 2017 to 2018, she was a postdoctoral scholar at the Computing and Mathematical Sciences department of the California Institute of Technology. She received her Ph.D. from the Systems and Control group in the Electrical Engineering department at the Eindhoven University of Technology in the Netherlands in 2017. She received her B.Sc. degree cum laude in Mechanical Engineering in 2010 at the Delft University of Technology. In 2012, she received her M.Sc. degree cum laude in Systems & Control at the Delft University of Technology, The Netherlands. Her research interests are in the identification, verification, and control of cyber-physical systems for temporal logic specifications and performance objectives.

Fei Chen is currently a Ph.D. student in the Division of Decision and Control Systems at KTH Royal Institute of Technology (Sweden). He received his M.Sc. degree from the Systems and Control group in the Electrical Engineering Department at Eindhoven University of Technology (Netherlands) in 2016. He received his B.Sc. degree in the Department of Control Science and Engineering at Zhejiang University (China) in 2014. His research interests are situated on the edge between control theory and formal methods in computer science, with particular interests in formal verification and control synthesis for multi-agent systems under temporal logic specifications.

Alessandro Abate (S'02–M'08–SM'19) is Professor of Verification and Control in the Department of Computer Science at the University of Oxford (UK), and is a fellow of the Alan Turing Institute in London (UK). He received a Laurea in Electrical Engineering in October 2002 from the University of Padova (IT), an MS in May 2004 and a PhD in December 2007, both in Electrical Engineering and Computer Sciences, at UC Berkeley (USA). He has been an International Fellow in the CS Lab at SRI International in Menlo Park (USA), and a PostDoctoral Researcher at Stanford University (USA), in the Department of Aeronautics and Astronautics. From June 2009 to mid 2013 he has been an Assistant Professor at the Delft Centre for Systems and Control, TU Delft - Delft University of Technology (NL).

Siep Weiland is Full Professor at the Control Systems Group, Dept. of Electrical Engineering, Eindhoven University of Technology. He received both his MSc. (1986) and PhD degrees in mathematics from the University of Groningen in the Netherlands. He was a postdoctoral research associate at the Dept. of Electrical Engineering and Computer Engineering, Rice University, Houston, USA from 1991 to 1992. Since 1992 he has been affiliated to Eindhoven University of Technology. His research interests are the general theory of systems and control, robust control, model approximation, modeling and control of hybrid systems, identification, and model predictive control.