Abstracting Partially Feedback Linearizable Systems Compositionally

Omar Hussien, Aaron Ames, and Paulo Tabuada

***Abstract*—Symbolic controller synthesis offers the ability to design controllers enforcing a rich class of specifications such as those expressible in temporal logic. Despite the promise of symbolic controller synthesis and correct-by-design control software, this design methodology is not yet widely applicable due to the complexity of constructing finite-state abstractions for large continuous systems. In this letter, we investigate a compositional approach to the construction of abstractions by exploiting the cascading structure of partially feedback linearizable systems. We show how the linearized part and the zero dynamics can be independently abstracted and subsequently composed to obtain an abstraction of the original continuous system. We also illustrate through examples how this compositional approach significantly reduces the time required for construction of abstractions.**

***Index Terms*—Hybrid systems, nonlinear output feedback.**

I. Introduction

THE VERIFICATION of control software for cyber-physical systems is gradually becoming more challenging given the increasing complexity of these systems. A recent approach to handle the verification problem is to synthesize control software using correct-by-design methods. These are techniques that synthesize both the control software as well as a proof of its correctness so that a-posteriori verification is greatly reduced or not even required. One of the widely used correct-by-design techniques is based on the construction of a finite-state abstraction for the given control system. A controller enforcing the specification can then be synthesized for the abstraction and subsequently lifted to a controller acting on the control system. Control software synthesis based on abstractions has two advantages over more traditional control design techniques: 1) it allows the use of more complex specifications such as those expressed in temporal logic; 2) controller synthesis is completely automated and consists of computing a fixed-point over the finite-state abstraction, which can always be done in finite time for finite-state abstractions [20]. The construction of abstractions for control systems that are incrementally input-to-state stable was presented in [13], [15], and [16]. Zamani *et al.* [23] showed that finite-state abstractions can still be computed even if incremental input-to-state stability fails to hold. Furthermore, various software tools for correct-by-design controller synthesis, using abstractions, have been developed and include PESSOA [7], CoSyMa [10], TuLiP [22], and SCOTS [19].

One of the drawbacks of abstraction based control software synthesis is that the computation of abstractions does not scale well with the number of states of the control system. Hence, it becomes infeasible to compute abstractions for large systems. One way to alleviate this problem is to avoid computing the abstraction monolithically, and compute it compositionally. Recent results for the compositional construction of abstractions were presented for discrete control systems in [14], linear systems in [12] and nonlinear control systems that can be decoupled into smaller subsystems using the same inputs in [17]. Similar results were introduced in [11] for a collection of identical decoupled switched systems subject to counting constraints and in [6] and [9] for nonlinear cooperative control systems in which the interaction between subsystems is modeled as a disturbance.

In this letter, we focus on control systems that are partially feedback linearizable [4]. We use this assumption to decompose such systems into its feedback linearizable part and its zero dynamics. This cascade decomposition can be exploited to compute abstractions compositionally: abstractions of the feedback linearizable part and of the zero dynamics are independently computed and then composed to obtain an abstraction of the original system.

By focusing on a different class of systems, the proposed compositional approach complements the compositional results reported in the literature. Moreover, since the class of partially feedback linearizable systems is reasonably large (e.g., automotive systems [2], drones [5] and bipedal robots [21]), the proposed results are quite useful in practice. To further substantiate this claim we present in Section V two examples. The first example is a truck and trailer system where we increase the number of trailers to illustrate how the proposed methodology scales better with the number of continuous states than the existing monolithic approach. The second example is a two-link model of a bipedal robot that is used to synthesize controllers providing a walking gait on a downward ramp. While it is important to move from computing abstractions compositionally to synthesize controllers compositionally, this is a much more challenging problem that is not addressed in this letter.

The remainder of this letter is organized as follows. Section II introduces the class of control systems we consider in this letter. In Section III we review the definition of

Manuscript received March 6, 2017; revised May 4, 2017; accepted May 27, 2017. Date of publication June 8, 2017; date of current version June 22, 2017. This work was supported by the NSF CPS under Award 1239085. Recommended by Senior Editor C. Prieur. *(Corresponding author: Omar Hussien.)*

O. Hussien and P. Tabuada are with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90066 USA (e-mail: ohussien@ucla.edu; tabuada@ee.ucla.edu).

A. Ames is with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ames@cds.caltech.edu).

Digital Object Identifier 10.1109/LCSYS.2017.2713461

different types of approximate simulation relations. Our main contribution appears in Section IV. We illustrate the benefits of our approach through different examples in Section V. This letter ends with several concluding remarks in Section VI.

II. Control Systems and Cascade Decompositions

A. Notation

We use $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{R}_0^+$ to denote the set of integer, real, positive and nonnegative real numbers. Given a vector $x \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{m\times n}$, we denote by $\|x\|$ and $\|A\|$ the infinity norm of x and A, respectively. We define the discretization of $S \subset \mathbb{R}^n$ by:

$$[S]_\alpha = \{s \in S | s_i = k_i\alpha, k_i \in \mathbb{Z}, i = 1, \ldots, n\},$$

where $\alpha \in \mathbb{R}^+$ is the discretization parameter. Given a measurable function $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}^n$, we denote the (essential) supremum (ess) $\sup_{t\in\mathbb{R}_0^+} \|f(t)\|$ by $\|f\|_\infty$. A continuous function $\gamma : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$; furthermore γ belongs to class $\mathcal{K}_\infty$ if $\gamma \in \mathcal{K}$ and $\gamma(r) \rightarrow \infty$ as $r \rightarrow \infty$. We use $1_{\mathbb{R}^n} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ to denote the identity map on $\mathbb{R}^n$.

B. Control Systems and Cascade Decompositions

In this letter we work with continuous time control systems defined as follows.

Definition 1: A control system $\Sigma = (\mathbb{R}^n, U, \mathcal{U}, f)$ consists of:

- the state space $\mathbb{R}^n$;
- the input set $U \subseteq \mathbb{R}^m$;
- the admissible input curves $\mathcal{U}$, a subset of all the piecewise continuous functions of time from intervals of the form $]a, b[\subset \mathbb{R}$ to U with $a < 0 < b$;
- the locally Lipschitz continuous map $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ defining the dynamics of the system.

We say Σ is a single-input control system when $m = 1$. We denote the trajectory of a control system Σ by $\xi_{x\upsilon}$:$]a, b[\rightarrow \mathbb{R}$ if there exists $\upsilon \in \mathcal{U}$ such that $\dot{\xi}_{x\upsilon} = f(\xi_{x\upsilon}, \upsilon)$. We also use the notation $\xi_{x\upsilon}(\tau)$ to denote the point reached by system Σ, at time τ, from the initial state x while applying the input υ. Note that this point is uniquely determined due to the Lipschitz continuity assumption on f [4]. A control system is forward complete if every trajectory is defined on an interval of the form $]a, \infty[$, where $a \in \mathbb{R}$. Necessary and sufficient conditions for forward completeness can be found in [1].

The results presented in this letter are proved for the class of control systems that can be defined as a cascade composition of smaller subsystems. We now present a definition of cascade decomposition tailored to the decompositions that arise from single-input partially feedback linearizable systems.

Definition 2: Let $\Sigma = (\mathbb{R}^n, U, \mathcal{U}, f)$ be a control system, let $x = (z, w) \in \mathbb{R}^n$, where $z = (x_1, \ldots, x_p)$ and $w = (x_{p+1}, \ldots, x_n)$ for some $p \leq n$, and let $v = (z, u)$ for $u \in U \subseteq \mathbb{R}$. System Σ admits a cascade decomposition into $\Sigma_1 = (\mathbb{R}^p, U, \mathcal{U}_1, f_1)$ and $\Sigma_2 = (\mathbb{R}^{n-p}, \mathbb{R}^{p+1}, \mathcal{U}_2, g)$ if:

$$\begin{aligned} f(x, u) &= (f_1(z, u), f_2(w, v)) \\ f_1(z, u) &= \big(x_2, x_3, \ldots, x_p, u\big) \\ f_2(w, v) &= g(w, v). \end{aligned}$$

Accordingly, system Σ can be seen as a cascade composition of Σ_1, of the form $\dot{z} = f_1(z, u)$, and Σ_2, of the form $\dot{w} = g(w, v)$, where the input of Σ_2 is connected to the output of Σ_1 according to $v = (z, u)$. Given a single-input partially feedback linearizable system Σ of relative degree p, we can always decompose it into its feedback linearized part and the residual dynamics [4]. The feedback linearized part corresponds to subsystem Σ_1 while the residual dynamics corresponds to Σ_2. Note that although Definition 2 describes a decomposition of Σ into two subsystems, Σ_1 and Σ_2, all the following results are still valid when we have N subsystems, e.g., when we have more than one input in a partially feedback linearizable system. We use $\xi_{x\upsilon}$, $\xi_{z\upsilon}$, and $\xi_{w\nu}$ to denote the trajectories Σ, Σ_1 and Σ_2, respectively. Let $\pi_1 : \mathbb{R}^n \rightarrow \mathbb{R}^p$ and $\pi_2 : \mathbb{R}^n \rightarrow \mathbb{R}^{n-p}$ be the natural projections on the first p and last $n-p$ entries, respectively. Rather than writing $\pi_1 \circ \xi_{x\upsilon}$ and $\pi_2 \circ \xi_{x\upsilon}$ we use the simpler notation $\xi^1_{x\upsilon}$ and $\xi^2_{x\upsilon}$, respectively. Note that whenever the input curves are assumed to be constant, we will use u and v instead of the Greek letters υ and ν, respectively.

C. Divergence of Trajectories

To prove the existence of different types of simulation relations between abstractions and control systems, we need to define a bound on the divergence of trajectories. This is captured by the notion of incremental forward completeness [23].

Definition 3: A control system Σ is incrementally forward complete (δ-FC) if it is forward complete and there exist continuous functions $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ and $\gamma : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ such that for every $s \in \mathbb{R}^+$, the functions $\beta(\cdot, s)$ and $\gamma(\cdot, s)$ belong to class $\mathcal{K}_\infty$ and for any $x, x' \in \mathbb{R}^n$, any $\tau \in \mathbb{R}^+$ and any $\upsilon, \upsilon' : [0, \tau[\rightarrow \mathbb{R}$, the following condition is satisfied for all $t \in [0, \tau[$:

$$\|\xi_{x\upsilon}(t) - \xi_{x'\upsilon'}(t)\| \leq \beta\big(\|x - x'\|, t\big) + \gamma\big(\|\upsilon - \upsilon'\|_\infty, t\big).$$

In other words, a control system is incrementally forward complete if the distance between any two trajectories starting from different initial states while applying different inputs for the same duration of time can be bounded by the functions β and γ that depends on the difference between the initial states and the difference between the inputs, respectively.

III. Symbolic Models and Approximate Simulation Relations

A. Systems

We briefly introduce the notion of system which will be used later to model all the systems of interest. Further details on the notion of system can be found in [20].

Definition 4: A system S is a quintuple $(X, U, \longrightarrow, Y, H)$ consisting of:

- A set of states X;
- A set of inputs U;
- A transition relation $\longrightarrow \subseteq X \times U \times X$;
- An output set Y;
- An output map $H : X \rightarrow Y$.

A system S is metric, if there exists a metric $d : Y \times Y \rightarrow \mathbb{R}_0^+$. We call state x' a u-successor for state x if the transition $x \xrightarrow{u} x'$ exists in the system. We also introduce the set of u-successors of a state x, denoted by $\mathbf{Post}_u(x)$, as well as the set of inputs $u \in U$, denoted by $U(x)$, such that $\mathbf{Post}_u(x)$ is nonempty.

B. Simulation Relations

We now introduce different types of simulation relations which we use to relate the computed abstractions to the control system of interest. We start with the notion of approximate simulation relation [3].

Definition 5: Let $S_a = (X_a, U_a, \underset{a}{\longrightarrow}, Y_a, H_a)$ and $S_b = (X_b, U_b, \underset{b}{\longrightarrow}, Y_b, H_b)$ be metric systems with the same output sets $Y_a = Y_b$ and metric d, and consider a precision $\varepsilon \in \mathbb{R}^+$. A relation $R \subseteq X_a \times X_b$ is said to be an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied:

1) for every $x_a \in X_a$, there exists $x_b \in X_b$ with $(x_a, x_b) \in R$;
2) for every $(x_a, x_b) \in R$ we have $d(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3) for every $(x_a, x_b) \in R$ we have that $x_a \xrightarrow{u_a} x'_a$ in S_a implies the existence of $x_b \xrightarrow{u_b} x'_b$ in S_b satisfying $(x'_a, x'_b) \in R$.

We denote the existence of an ε-approximate simulation relation from S_a to S_b by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$.

While simulation relations are useful for verification purposes, when the objective is the synthesis of controllers, the relevant notion is alternating simulation. See [20] for a comparison between these two different, but related, notions.

Definition 6: Let $S_a = (X_a, U_a, \xrightarrow[a]{}, Y_a, H_a)$ and $S_b = (X_b, U_b, \xrightarrow[b]{}, Y_b, H_b)$ be metric systems with the same output sets $Y_a = Y_b$ and metric d, and consider a precision $\varepsilon \in \mathbb{R}^+$. A relation $R \subseteq X_a \times X_b$ is said to be an ε-approximate alternating simulation relation from S_a to S_b if the first two conditions in Definition 5 and the following condition are satisfied:

- for every $(x_a, x_b) \in R$ and for every $u_a \in U_a(x_a)$ there exists $u_b \in U_b(x_b)$ such that for every $x'_b \in \mathbf{Post}_{u_b}(x_b)$ there exists $x'_a \in \mathbf{Post}_{u_a}(x_a)$ satisfying $(x'_a, x'_b) \in R$.

We denote the existence of an ε-approximate alternating simulation relation from S_a to S_b by $S_a \preceq_{\mathcal{AS}}^{\varepsilon} S_b$.

Note that the existence of these simulation relations enables the refinement of controllers synthesized for the abstractions to controllers that act on the control system [20].

C. Symbolic Models

We define the time-discretization of a control system Σ, denoted by $S_\tau(\Sigma)$, where $\tau \in \mathbb{R}^+$ is the sampling time, as follows:

$$S_\tau(\Sigma) = \left(\mathbb{R}^n, U_\tau, \xrightarrow[\tau]{}, \mathbb{R}^n, 1_{\mathbb{R}^n}\right), \tag{1}$$

where:

- $U_\tau = \{u : [0, \tau[\to U | u(t) = u(0), t \in [0, \tau[\}$;
- $x_\tau \xrightarrow{u_\tau} x'_\tau$ if $\xi_{x_\tau u_\tau}(\tau) = x'_\tau$,

for $x_\tau, x'_\tau \in \mathbb{R}^n$ and $u_\tau \in U_\tau$.

We compute an abstraction of a control system by discretizing the states, the inputs and the time.

Definition 7: Given the control system $\Sigma = (\mathbb{R}^n, U, \mathcal{U}, f)$, a map $\delta : (\mathbb{R}^+)^4 \to \mathbb{R}^+$, and the triple $q = (\tau, \eta, \mu)$ of quantization parameters, where $\tau \in \mathbb{R}^+$ is the sampling time, $\eta \in \mathbb{R}^+$ is the state space quantization, and $\mu \in \mathbb{R}^+$ is the input quantization, the abstraction of Σ associated with q and δ is the system $S_{q\delta}(\Sigma) = (X, U, \longrightarrow, Y, H)$ defined by:

- $X = [\mathbb{R}^n]_\eta$;
- $U = [\mathbb{R}^m]_\mu$;
- $x \xrightarrow{u} x'$ if $\|\xi_{xu}(\tau) - x'\| \leq \delta(\varepsilon, \tau, \eta, \mu)$;
- $Y = \mathbb{R}^n$;
- $H = \imath : X \hookrightarrow Y$,

where $\varepsilon \in \mathbb{R}^+$ and $\imath$ is the natural embedding of X into Y. Note that, an abstraction is finite if it has a finite set of states and a finite set of inputs, which can be achieved if the state space and the input space are restricted to bounded sets.

Given a system Σ that admits a cascade decomposition into Σ_1 and Σ_2, as in Definition 2, instead of Σ_2 we work with the system:

$$\tilde{\Sigma}_2 = (\mathbb{R}^{n-p}, \mathbb{R}^{p+1}, \mathcal{U}_2, g(w, \xi_{zu}(t), u)) \tag{2}$$

where $\mathcal{U}_2$ is the set of constant curves and we regard $\dot{w} = g(w, \xi_{zu}(t), u)$ as a *time-varying* differential equation with constant input (z, u).

Definition 8: Let the abstractions of Σ_1 and $\tilde{\Sigma}_2$ be $S_{q_1\delta_1}(\Sigma_1) = (X_1, U_1, \xrightarrow[1]{}, Y_1, H_1)$ and $S_{q_2\delta_2}(\tilde{\Sigma}_2) = (X_2, U_2, \xrightarrow[2]{}, Y_2, H_2)$, respectively where:

$$q_1 = (\tau, \eta_1, \eta_1), \quad q_2 = (\tau, \eta_2, \eta_1), \quad \tau, \eta_1, \eta_2 \in \mathbb{R}^+,$$

The composed abstraction of Σ denoted by:

$$S_{q_1\delta_1}(\Sigma_1) \circledcirc S_{q_2\delta_2}(\tilde{\Sigma}_2) = \left(X, U, \longrightarrow, Y, H_{q_1\delta_1 q_2\delta_2}\right)$$

is defined by:

- $X = X_1 \times X_2$;
- $U = U_1$;
- $(x_1, x_2) \xrightarrow{u_1} (x'_1, x'_2)$ if $x_1 \xrightarrow{u_1} x'_1$ in $S_{q_1\delta_1}(\Sigma_1)$, $x_2 \xrightarrow{u_2} x'_2$ in $S_{q_2\delta_2}(\tilde{\Sigma}_2)$ and $u_2 = (x_1, u_1)$;
- $Y = \mathbb{R}^n = Y_1 \times Y_2$;
- $H_{q_1\delta_1 q_2\delta_2} = (\imath_1, \imath_2) : X_1 \times X_2 \hookrightarrow Y_1 \times Y_2$.

IV. Symbolic Models for δ-FC Cascade Control Systems

In this section we present our main result. Given a control system Σ that admits a cascade decomposition as in Definition 2, we prove the existence of different types of simulation relations between the control system and the abstraction obtained by composing the abstractions of the subsystems as in Definition 8.

Theorem 1: Let Σ be a control system that can be decomposed into Σ_1 and Σ_2, as in Definition 2, and let $\tilde{\Sigma}_2$ be the system defined in (2). Let $S_{q_1\delta_1}(\Sigma_1) \circledcirc S_{q_2\delta_2}(\tilde{\Sigma}_2)$ be the composed abstraction given in Definition 8 and consider any precision $\varepsilon \in \mathbb{R}^+$. Under the following assumptions:

- $\max\{\eta_1, \eta_2\} \leq \varepsilon$;
- Σ_1 and $\tilde{\Sigma}_2$ are δ-FC control systems;
- $\delta_1(\varepsilon, \tau, \eta_1, \eta_1) = \beta_1(\varepsilon, \tau) + \eta_1$;
- $\delta_2(\varepsilon, \tau, \eta_2, \eta_1) = \beta_2(\varepsilon, \tau) + \gamma_2(\varepsilon, \tau) + \eta_2$,

where β_i and γ_i for $i = 1, 2$ are the functions in Definition 3, we have:

$$S_{q_1\delta_1}(\Sigma_1) \circledcirc S_{q_2\delta_2}(\tilde{\Sigma}_2) \preceq_{\mathcal{AS}}^{\varepsilon} S_\tau(\Sigma). \tag{3}$$

Instead, if δ_1 is defined by:

$$\delta_1(\varepsilon, \tau, \eta_1, \eta_1) = \beta_1(\varepsilon, \tau) + \gamma_1(\eta_1, \tau) + \eta_1,$$

we also have:

$$S_\tau(\Sigma) \preceq_{\mathcal{S}}^{\varepsilon} S_{q_1\delta_1}(\Sigma_1) \circledcirc S_{q_2\delta_2}(\tilde{\Sigma}_2). \tag{4}$$

Proof: **First** we prove (3). Consider the relation $R \subseteq X \times \mathbb{R}^n$ defined by $((x_1, x_2), (z, w)) \in R$ iff:

$$d(H_{q_1\delta_1 q_2\delta_2}(x_1, x_2), H(z, w)) = \|(x_1, x_2) - (z, w)\| \leq \varepsilon.$$

By choosing $z = x_1$ and $w = x_2$, $((x_1, x_2), (z, w)) \in R$ and conditions (1-2) in Definition 6 are satisfied. Now we show that condition (3) in Definition 6 is satisfied for every $((x_1, x_2), (z, w)) \in R$. Consider any $u_1 \in U_1$ and let $u \in U_\tau$ be equal to u_1. Consider the unique transition $(z, w) \xrightarrow{u} (z', w') = \xi_{xu}(\tau) \in \mathbf{Post}_u(z, w)$ in $S_\tau(\Sigma)$. To prove the existence of a transition in $S_{q_1\delta_1}(\Sigma_1) \circledcirc S_{q_2\delta_2}(\tilde{\Sigma}_2)$ we need to show that: (i) $x_1 \xrightarrow{u_1} x'_1$ in $S_{q_1\delta_1}(\Sigma_1)$, (ii) $x_2 \xrightarrow{u_2} x'_2$ in $S_{q_2\delta_2}(\tilde{\Sigma}_2)$, and (iii) $u_2 = (x_1, u_1)$ hold.

Since for all $((x_1, x_2), (z, w)) \in R$, $\|(x_1, x_2) - (z, w)\| \leq \varepsilon$ and as we are using the infinity norm, we obtain:

$$\max\{\|x_1 - z\|, \|x_2 - w\|\} = \|(x_1, x_2) - (z, w)\| \leq \varepsilon. \tag{5}$$

We start by proving (i) as follows. Consider $x'_1 = [\xi_{zu}(\tau)]_{\eta_1}$, we have:

$$\left\|\xi_{xu}^1(\tau) - x'_1\right\| = \left\|\xi_{xu}^1(\tau) - \left[\xi_{zu}(\tau)\right]_{\eta_1}\right\| \leq \eta_1. \tag{6}$$

Given that Σ_1 is δ-FC, $u = u_1$ and using (5)-(6), we have:

$$\begin{aligned} \left\|\xi_{x_1u_1}(\tau) - x'_1\right\| &\leq \left\|\xi_{x_1u_1}(\tau) - \xi_{xu}^1(\tau)\right\| + \left\|\xi_{xu}^1(\tau) - x'_1\right\| \\ &\leq \beta_1(\|x_1 - z\|, \tau) + \eta_1 \\ &\leq \beta_1(\varepsilon, \tau) + \eta_1, \end{aligned} \tag{7}$$

which implies the existence of $x_1 \xrightarrow{u_1} x_1'$ in $S_{q_1\delta_1}(\Sigma_1)$.

Now we show that (ii) and (iii) hold. Consider $x_2' = [\xi_{wv}(\tau)]_{\eta_2}$, we obtain:

$$\left\| \xi^2_{xu}(\tau) - x_2' \right\| = \left\| \xi^2_{xu}(\tau) - [\xi_{wv}(\tau)]_{\eta_2} \right\| \leq \eta_2. \tag{8}$$

Given that $\tilde{\Sigma}_2$ is δ-FC, $u_2 = (x_1, u_1) = (x_1, u)$ and using (5) and (8), we have:

$$\begin{aligned} \left\| \xi_{x_2u_2}(\tau) - x_2' \right\| &\leq \left\| \xi_{x_2u_2}(\tau) - \xi^2_{xu}(\tau) \right\| + \left\| \xi^2_{xu}(\tau) - x_2' \right\| \\ &\leq \beta_2(\|w - x_2\|, \tau) + \gamma_2(\|u_2 - v\|, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\|u_2 - v\|, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\|x_1 - z\|, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\varepsilon, \tau) + \eta_2, \end{aligned} \tag{9}$$

which implies the existence of $x_2 \xrightarrow{u_2} x_2'$ in $S_{q_2\delta_2}(\tilde{\Sigma}_2)$.

From (7), (9) and $u_2 = (x_1, u_1)$, we conclude the existence of $(x_1, x_2) \xrightarrow{u} (x_1', x_2')$ in the composed system $S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2)$. Using (6) and (8), we obtain:

$$\begin{aligned} \left\| (x_1', x_2') - (z', w') \right\| &= \max\left\{ \|x_1' - z'\|, \|x_2' - w'\| \right\} \\ &= \max\{\eta_1, \eta_2\} \leq \varepsilon, \end{aligned} \tag{10}$$

which implies that $((x_1', x_2'), (z', w'))$ belongs to R, hence $S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2) \preceq^{\varepsilon}_{\mathcal{AS}} S_\tau(\Sigma)$.

Second we prove (4). Consider the relation $R \subseteq \mathbb{R}^n \times X$ defined by $((z, w), (x_1, x_2)) \in R$ iff:

$$d(H(z, w), H_{q_1\delta_1q_2\delta_2}(x_1, x_2)) = \|(z, w) - (x_1, x_2)\| \leq \varepsilon.$$

Since for all $(z, w) \in \mathbb{R}^n$, there exists a $(x_1, x_2) \in X$ satisfying:

$$\begin{aligned} \|(z, w) - (x_1, x_2)\| &= \max\{\|z - x_1\|, \|w - x_2\|\} \\ &= \max\{\eta_1, \eta_2\} \leq \varepsilon, \end{aligned} \tag{11}$$

as we are using the infinity norm, hence $((z, w), (x_1, x_2)) \in R$ and conditions (1-2) in Definition 5 are satisfied. Now we show that condition (3) in Definition 5 is satisfied for every $((z, w), (x_1, x_2)) \in R$. Consider any $u \in U_\tau$ we pick u_1 such that:

$$\|u - u_1\| \leq \eta_1. \tag{12}$$

Consider the transition $(z, w) \xrightarrow{u} (z', w') = \xi_{xu}(\tau)$ in $S_\tau(\Sigma)$. To prove the existence of a transition in $S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2)$ we need to show that: (i) $x_1 \xrightarrow{u_1} x_1'$ in $S_{q_1\delta_1}(\Sigma_1)$, (ii) $x_2 \xrightarrow{u_2} x_2'$ in $S_{q_2\delta_2}(\tilde{\Sigma}_2)$, and (iii) $u_2 = (x_1, u_1)$ hold.

We start by proving (i) as follows. Consider $x_1' = [\xi_{zu}(\tau)]_{\eta_1}$, we obtain:

$$\left\| \xi^1_{xu}(\tau) - x_1' \right\| = \left\| \xi^1_{xu}(\tau) - [\xi_{zu}(\tau)]_{\eta_1} \right\| \leq \eta_1. \tag{13}$$

Given that Σ_1 is δ-FC, and using (11)-(13), we have:

$$\begin{aligned} \left\| \xi_{x_1u_1}(\tau) - x_1' \right\| &\leq \left\| \xi_{x_1u_1}(\tau) - \xi^1_{xu}(\tau) \right\| + \left\| \xi^1_{xu}(\tau) - x_1' \right\| \\ &\leq \beta_1(\|x_1 - z\|, \tau) + \gamma_1(\|u_1 - u\|, \tau) + \eta_1 \\ &\leq \beta_1(\varepsilon, \tau) + \gamma_1(\|u_1 - u\|, \tau) + \eta_1 \\ &\leq \beta_1(\varepsilon, \tau) + \gamma_1(\eta_1, \tau) + \eta_1, \end{aligned} \tag{14}$$

which implies the existence of $x_1 \xrightarrow{u_1} x_1'$ in $S_{q_1\delta_1}(\Sigma_1)$.

Now we show that (ii) and (iii) hold. Consider $x_2' = [\xi^2_{xu}(\tau)]_{\eta_2}$, we obtain:

$$\left\| \xi^2_{xu}(\tau) - x_2' \right\| = \left\| \xi^2_{xu}(\tau) - [\xi_{wv}(\tau)]_{\eta_2} \right\| \leq \eta_2. \tag{15}$$

Given that $\tilde{\Sigma}_2$ is δ-FC, $u_2 = (x_1, u_1)$ and using (11), (12) and (15), we have:

$$\begin{aligned} \left\| \xi_{x_2u_2}(\tau) - x_2' \right\| &\leq \left\| \xi_{x_2u_2}(\tau) - \xi^2_{xu}(\tau) \right\| + \left\| \xi^2_{xu}(\tau) - x_2' \right\| \\ &\leq \beta_2(\|w - x_2\|, \tau) + \gamma_2(\|u_2 - v\|, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\|(x_1, u_1) - v\|, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\max\{\varepsilon, \eta_1\}, \tau) + \eta_2 \\ &\leq \beta_2(\varepsilon, \tau) + \gamma_2(\varepsilon, \tau) + \eta_2, \end{aligned} \tag{16}$$

which implies the existence of $x_2 \xrightarrow{u_2} x_2'$ in $S_{q_2\delta_2}(\tilde{\Sigma}_2)$.

From (14), (16) and $u_2 = (x_1, u_1)$, we conclude the existence of $(x_1, x_2) \xrightarrow{u} (x_1', x_2')$ in the composed system $S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2)$. Using (13) and (15) we obtain:

$$\begin{aligned} \left\| (z', w') - (x_1', x_2') \right\| &= \max\left\{ \|z' - x_1'\|, \|w' - x_2'\| \right\} \\ &= \max\{\eta_1, \eta_2\} \leq \varepsilon, \end{aligned} \tag{17}$$

which implies that $((z', w'), (x_1', x_2')) \in R$, hence $S_\tau(\Sigma) \preceq^{\varepsilon}_{\mathcal{S}} S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2)$. ■

Remark 1: When (3) holds, a controller synthesized for the abstraction can be refined to a controller for the original control system enforcing the same specification [20]. However, non-existence of a controller for the abstraction does not imply non-existence of a controller for the original control system. Whenever a controller cannot be found on the abstraction, a more detailed abstraction can be computed using smaller discretization parameters until a controller is found or the discretization parameters are considered to be too small.

Remark 2: Although (3) and (4) are the same inequalities that appear in [23, Th. 4.1], our approach results in a more conservative abstraction, i.e., has more transitions. Accordingly, there might be controllers for a monolithic abstraction that cannot be found when working with $S_{q_1\delta_1}(\Sigma_1) \circledgtr S_{q_2\delta_2}(\tilde{\Sigma}_2)$. We return to this point in Section V in the context of a specific example. Under stronger stability assumptions, a version of Theorem 1 guaranteeing the existence of an approximate bisimulation between the abstraction and the concrete system can be proved using techniques similar to those used to prove Theorem 1. When an approximate bisimulation exists, the non-existence of a controller for the abstraction implies the non-existence of a controller for the concrete system.

V. Experimental Results

In this section we illustrate our results on two examples. First, we compare our results to the monolithic approach using a truck and trailer system, similar to the example considered in [18]. We show how the proposed compositional abstraction technique scales better, as the number of trailers increases, than the monolithic approach. In the second example we synthesize a controller, using an abstraction computed with the proposed compositional approach, for the two-link biped robot, also known as the compass biped, which appeared in [21, Sec. 3.4.6]. All the computations were done on a 3.4 GHz iMac with 32GB of RAM.

A. Truck and Trailer Example

Consider a truck connected to n trailers by a spring-damper system, shown in Fig. 1, which can be modeled by:

$$\begin{aligned} \dot{d}_1 &= v_2 - v_1, \\ \dot{v}_1 &= \frac{K_s}{m} d_1 + \frac{K_d}{m}(v_2 - v_1), \\ \dot{d}_2 &= v_3 - v_2, \\ \dot{v}_2 &= \frac{K_s}{m} d_2 + \frac{K_d}{m}(v_3 - v_2), \\ &\vdots \\ \dot{v}_{n+1} &= u, \end{aligned} \tag{18}$$

where d_i is the distance between trailers i and $i+1$, and v_i is the velocity of trailer i, for $i = 1, \ldots, n$. The spring-damper constants are denoted by K_s and K_d, m is the mass of the

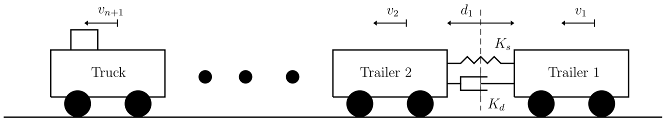


Fig. 1. Truck and trailers system.

TABLE I
TIME SPENT TO COMPUTE ABSTRACTIONS, FOR DIFFERENT NUMBER OF TRAILERS, USING THE COMPOSITIONAL APPROACH AND THE MONOLITHIC APPROACH

Number of trailers	1	2	3
Compositional approach	10[sec]	700[sec]	6[hr]
Monolithic approach	25[sec]	2000[sec]	>24[hr]

trailer, u is the external acceleration input acting on the truck and v_{n+1} is the velocity of the truck, respectively.

We can regard the system described by (18) as a cascade composition of system Σ_1 given by:

$$\begin{aligned} \dot{d}_2 &= v_3 - v_2, \\ \dot{v}_2 &= \frac{K_s}{m} d_2 + \frac{K_d}{m}(v_3 - v_2), \\ &\vdots \\ \dot{v}_{n+1} &= u, \end{aligned} \tag{19}$$

and system Σ_2:

$$\begin{aligned} \dot{d} &= v_2 - v_1, \\ \dot{v}_1 &= \frac{K_s}{m} d + \frac{K_d}{m}(v_2 - v_1). \end{aligned} \tag{20}$$

Note that Σ is partially feedback linearizable since it is a linear system. However, our approach only relies on the ability of rendering the partially feedback linearizable part linear which is already the case in (19). Hence, we directly abstract (19) without designing a preliminary feedback rendering it a chain of integrators. This illustrates that our results are more general than the specific technical statement in Theorem 1.

We computed abstractions of system Σ_1 and Σ_2, using the MATLAB toolbox PESSOA [7], for different numbers of trailers. The state space and input space discretization parameters used were $\eta = 1$ and $\mu = 1$, respectively, whereas we used $\tau = 0.5$ for the sampling time. A comparison of the time spent to construct the abstraction of the full system, for 1, 2, and 3 trailers, using the proposed compositional approach and the traditional monolithic approach is listed in Table I. Note that the addition of each trailer increases the number of continuous states by 2. As the number of trailers increases, we observe in Table I a speedup by a factor of 4 in the time required to compute the abstraction when we have 3 trailers. Note also that only the relative time is of significance since the implementation of PESSOA is now several years old and can be optimized in several different ways.

B. Compass Biped Robot Example

Consider the compass biped robot model [21], shown in Fig. 2, and given by:

$$\begin{aligned} \ddot{q}_1 &= v, \\ \dot{q}_2 &= \frac{\sigma_2}{D_{2,2}(q_1)} - \frac{D_{2,1}(q_1)}{D_{2,2}(q_1)}\dot{q}_1, \\ \dot{\sigma}_2 &= -G_2(q_1, q_2, \alpha), \end{aligned} \tag{21}$$

where q_1 is the angle between the two legs, q_2 is the angle between the stance leg and the vertical to the ground, σ_2 is the momentum conjugate to q_2, v is the actuator torque applied at the joint between the two legs of the robot, α is the ground slope, and $D(q_1)$ and $G(q_1, q_2, \alpha)$ are the inertia matrix and the gravity vector obtained from [21, eqs. (3.58) and (3.60)], respectively. The parameters for this model were taken from [21, Table 3.1].

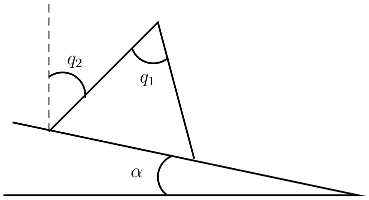


Fig. 2. Illustration of a compass bipedal robot over sloped ground.

Equation (21) describes the motion of a biped while one of the feet is above ground. To complete the model we need to describe what happens when a foot strikes the ground. We model this phenomenon by reset map in a hybrid automaton with a single mode. The foot strikes the ground whenever:

$$q_1 = 2q_2. \tag{22}$$

Upon this event, the role of the stance and swing legs is reversed and this is captured by an instantaneous change in the states described by the reset maps:

$$\begin{bmatrix} q_1^+ \\ q_2^+ \end{bmatrix} = \Delta_q \begin{bmatrix} q_1^- \\ q_2^- \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \dot{q}_1^+ \\ \sigma_2^+ \end{bmatrix} = \Delta_{\dot{q}}(q) \begin{bmatrix} \dot{q}_1^- \\ \sigma_2^- \end{bmatrix} \tag{23}$$

defined by [21, eqs. (3.54) and (3.56)].

Although Theorem 1 is not stated for hybrid systems, it can be applied to the hybrid system modeling a bipedal robot by first (compositionally) computing an abstraction of the continuous dynamics on its domain and then changing the abstraction to account for the effect of the reset map. We can regard the system described by (21) as a cascade composition of system Σ_1 given by:

$$\ddot{q}_1 = v, \tag{24}$$

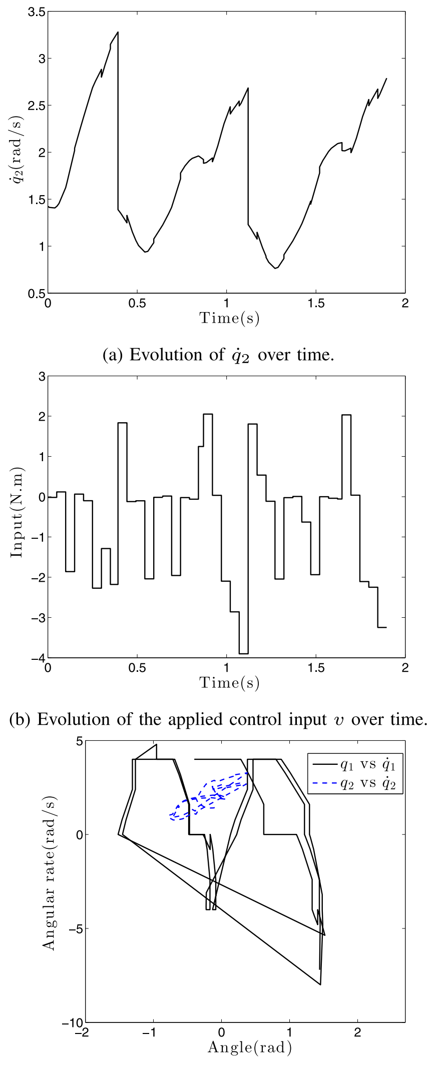
and system Σ_2:

$$\begin{aligned} \dot{q}_2 &= \frac{\sigma_2}{D_{2,2}(q_1)} - \frac{D_{2,1}(q_1)}{D_{2,2}(q_1)}\dot{q}_1, \\ \dot{\sigma}_2 &= -G_2(q_1, q_2, \alpha). \end{aligned} \tag{25}$$

We computed abstractions of system Σ_1 and $\tilde{\Sigma}_2$ using the MATLAB toolbox PESSOA [7]. For a desired precision $\varepsilon = 0.05$, the used state space and input space discretization parameters were $\eta = 0.05$ and $\mu = 0.05$, respectively, whereas we used $\tau = 0.05$ for the sampling time. The abstractions of Σ_1 and Σ_2 were computed in 20 seconds and 100 minutes, respectively, while composing them took 20 minutes. This resulted in 120 minutes to compute an abstraction compositionally. Constructing an abstraction for the full model monolithically, using the same discretization parameters, took 350 minutes. Hence, the proposed compositional approach was three times faster in this example.

In order to force the robot to move forward, $\dot{q}_2$ needs to be always greater than zero. Hence, we synthesized a controller that enforces $\dot{q}_2$ to be always greater than ε, i.e., greater than zero plus the precision of the abstraction. Fig. 3 shows the closed-loop simulation results and the phase portrait for the compass bipedal robot. The phase portrait indicates that non-periodic walking is achieved thereby illustrating the difference with existing design methods [8], [21] that produce periodic gaits.

We also synthesized a controller for the same specification using the monolithic abstraction. In order to illustrate that compositional abstractions can be conservative, we compare in Figure 4 the number of inputs available to enforce the



(a) Evolution of $\dot{q}_2$ over time.

(b) Evolution of the applied control input v over time.

(c) Phase portrait of the closed-loop system.

Fig. 3. Closed-loop simulation using the synthesized controller.

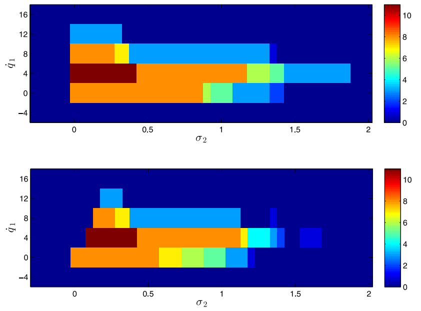


Fig. 4. Number of inputs available to enforce the specification at the cross section $q_1 = 0.4$ rad and $q_2 = 0.2$ rad using the monolithic abstraction (top) and using the compositional abstraction (bottom).

specification for a wide range of values of $\dot{q}_1$ and σ_2 when $q_1 = 0.4$ and $q_2 = 0.2$ rad. We can observe in Figure 4 that the controller synthesized using the monolithic abstraction is more permissive than the controller synthesized using our approach. However, the reduction in the available inputs is not substantial and thus only has a marginal effect in the ability to control the robot.

VI. Conclusion

In this letter, we presented a compositional approach to compute abstractions of continuous time control systems that admit cascade decompositions into smaller subsystems arising from partially feedback linearizability. Although the compositional approach is more conservative than the monolithic approach, it leads to a considerable speedup in the computation time. Using the truck and trailers system, we illustrated that using the proposed approach we could reduce the time required to compute an abstraction by a factor of 4 when compared with the monolithic approach. For the biped robot example, the computation time was reduced by a factor of 3. Moreover, by comparing the controller synthesized for the compositional abstraction with the controller synthesized for the monolithic abstraction we observed that the conservativeness of the compositional abstraction was not substantial.

References

[1] D. Angeli and E. D. Sontag, “Forward completeness, unboundedness observability, and their Lyapunov characterizations,” *Syst. Control Lett.*, vol. 38, nos. 4–5, pp. 209–217, 1999.

[2] L. Eriksson and L. Nielsen, *Modeling and Control of Engines and Drivelines*. Chichester, U.K.: Wiley, 2014.

[3] A. Girard and G. J. Pappas, “Approximation metrics for discrete and continuous systems,” *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 782–798, May 2007.

[4] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[5] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[6] R. Majumdar, K. Mallik, and A.-K. Schmuck, “Compositional synthesis of finite state abstractions,” *arXiv preprint arXiv:1612.08515*, 2016.

[7] M. Mazo, Jr., A. Davitian, and P. Tabuada, “Pessoa: A tool for embedded controller synthesis,” in *Computer Aided Verification*. Heidelberg, Germany: Springer, 2010, pp. 566–569.

[8] T. McGeer, “Passive dynamic walking,” *Int. J. Robot. Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[9] P.-J. Meyer, A. Girard, and E. Witrant, “Safety control with performance guarantees of cooperative systems using compositional abstractions,” *IFAC PapersOnLine*, vol. 48, no. 27, pp. 317–322, 2015.

[10] S. Mouelhi, A. Girard, and G. Gössler, “CoSyMA: A tool for controller synthesis using multi-scale abstractions,” in *Proc. 16th Int. Conf. Hybrid Syst. Comput. Control*, Philadelphia, PA, USA, 2013, pp. 83–88.

[11] P. Nilsson and N. Ozay, “Control synthesis for large collections of systems with mode-counting constraints,” in *Proc. 19th Int. Conf. Hybrid Syst. Comput. Control*, Vienna, Austria, 2016, pp. 205–214.

[12] P. Nilsson and N. Ozay, “Synthesis of separable controlled invariant sets for modular local control design,” in *Proc. Amer. Control Conf. (ACC)*, Boston, MA, USA, 2016, pp. 5656–5663.

[13] G. Pola, A. Girard, and P. Tabuada, “Approximately bisimilar symbolic models for nonlinear control systems,” *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[14] G. Pola, P. Pepe, and M. D. Di Benedetto, “Symbolic models for networks of control systems,” *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3663–3668, Nov. 2016.

[15] G. Pola, P. Pepe, M. D. Di Benedetto, and P. Tabuada, “Symbolic models for nonlinear time-delay systems using approximate bisimulations,” *Syst. Control Lett.*, vol. 59, no. 6, pp. 365–373, 2010.

[16] G. Pola and P. Tabuada, “Symbolic models for nonlinear control systems: Alternating approximate bisimulations,” *SIAM J. Control Optim.*, vol. 48, no. 2, pp. 719–733, 2009.

[17] G. Reißig, “Abstraction based solution of complex attainability problems for decomposable continuous plants,” in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Atlanta, GA, USA, 2010, pp. 5911–5917.

[18] M. Rungger, M. Mazo, Jr., and P. Tabuada, “Specification-guided controller synthesis for linear systems and safe linear-time temporal logic,” in *Proc. 16th Int. Conf. Hybrid Syst. Comput. Control*, Philadelphia, PA, USA, 2013, pp. 333–342.

[19] M. Rungger and M. Zamani, “Scots: A tool for the synthesis of symbolic controllers,” in *Proc. 19th Int. Conf. Hybrid Syst. Comput. Control*, Vienna, Austria, 2016, pp. 99–104.

[20] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. New York, NY, USA: Springer, 2009.

[21] E. R. Westervelt *et al.*, *Feedback Control of Dynamic Bipedal Robot Locomotion*, vol. 28. Boca Raton, FL, USA: CRC Press, 2007.

[22] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, “TuLiP: A software toolbox for receding horizon temporal logic planning,” in *Proc. 14th Int. Conf. Hybrid Syst. Comput. Control*, Chicago, IL, USA, 2011, pp. 313–314.

[23] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, “Symbolic models for nonlinear control systems without stability assumptions,” *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1804–1809, Jul. 2012.