Int. J. Appl. Math. Comput. Sci., 2020, Vol. 30, No. 4, 629–639
DOI: 10.34768/amcs-2020-0046

ON DISTRIBUTED SYMBOLIC CONTROL OF INTERCONNECTED SYSTEMS UNDER PERSISTENCY SPECIFICATIONS

W. ALEJANDRO APAZA-PEREZ [a,*], CHRISTOPHE COMBASTEL [a], ALI ZOLGHADRI [a]

[a] IMS Laboratory
University of Bordeaux—CNRS
351 Cours de la Libération, 33405 Talence, France
e-mail: {wapazaperez,christophe.combastel,ali.zolghadri}@u-bordeaux.fr

This paper presents an abstraction-based technique to solve the problem of distributed controller design enforcing persistency specifications for interconnected systems. For each subsystem, controller synthesis is based on local distributed sensor information from other subsystems. An effective method is presented for quantification of such partial information in an abstraction in terms of level sets of Lyapunov-like ranking functions. The results are illustrated on a laboratory hydraulic system.

Keywords: symbolic controller synthesis, distributed controller, interconnected system.

1. Introduction

Symbolic models are abstract descriptions of continuous systems where each symbol corresponds to an aggregate of continuous states (Tabuada, 2009). Such abstractions are computed based on some behavioral relationships, e.g., approximate bi-simulation or its alternating version (Pola *et al.*, 2008; 2010; Tabuada, 2009; Reissing, 2011; Tazaki and Imura, 2012; Zamani *et al.*, 2012; 2014; Borri *et al.*, 2012; Majumdar and Zamani, 2012; Dallal and Tabuada, 2015; Girard *et al.*, 2016). Over the last two decades, the use of symbolic models for control design has spurred on substantial research efforts, among many others; see, e.g., the works of Nilsson (2017), Weber *et al.* (2017), Gruber *et al.* (2017) or Nilsson and Ozay (2020) and the references therein. They are also other symbolic contexts, e.g., symbolic computing, with applications in probabilistic and stochastic analysis (Kamiński, 2015). The main motivation has been handling complex heterogeneous systems that should satisfy complex specifications (Belta *et al.*, 2017). Software tools are now available for the computation of abstractions, for example, PESSOA (Mazo *et al.*, 2010), CoSyMa (Mouelhi *et al.*, 2013), TuLiP (Wongpiromsarn *et al.*, 2011), or SCOTS (Rungger and Zamani, 2016).

Despite considerable progress in symbolic control techniques, the curse of dimensionality still yields restrictions: Indeed, the number of symbolic states (respectively inputs) increases exponentially with respect to the state-space (respectively input-space) dimension. Distributed control approaches have been proposed in the literature to improve the scalability (Zhai *et al.*, 2013; Ge *et al.*, 2017; Borri *et al.*, 2019; Jabri *et al.*, 2020), as well as compositional methods for symbolic controller synthesis (Mayer and Dimarogonas, 2017; Saoud *et al.*, 2018; 2020; Coënt *et al.*, 2016; Meyer *et al.*, 2018; Pola *et al.*, 2018; Kim *et al.*, 2015; Nilsson and Ozay, 2020).

In this paper, we address the problem of distributed symbolic control for interconnected systems. The developments in the present paper are in the spirit of the work reported by Dallal and Tabuada (2015) to controller synthesis. As in that paper, this work considers the problem of enforcing a persistency specification of the form "reach a set of states P and remain there for all future time", which is denoted in linear temporal logic (LTL) by $\Diamond\Box P$, meaning "eventually always" (see also Girard *et al.*, 2016; Nilsson, 2017; Weber *et al.*, 2017; Gruber *et al.*, 2017; Nilsson and Ozay, 2020; Belta *et al.*, 2017). Unlike the approaches of the papers above, except Dallal and Tabuada (2015), the notion of ranking functions is used in this paper, which represents local distributed information for controller synthesis. The persistency

[*] Corresponding author

specification and ranking functions are very close in spirit to asymptotic stability and Lyapunov functions in classical control theory, and hence it gives an option to build a bridge between classical control theory and symbolic control methods.

Dallal and Tabuada (2015) presented a compositional approach has been presented for the design of controllers that enforce reach-and-stay specifications. The method is inspired by small gain results from control theory (Jiang *et al.*, 1994; Dashkovskiy *et al.*, 2010) and assume-guarantee contracts for dynamical systems (Saoud *et al.*, 2019) based on prior works from formal methods; see the work of Henzinger *et al.* (2002) and the references therein. In this paper, we extend the basic results reported by Dallal and Tabuada (2015) for 2-component systems to n-dimensional interconnected nondeterministic systems, which was partially presented by Apaza-Perez *et al.* (2019). Unlike Dallal and Tabuada (2015) or Apaza-Perez *et al.* (2019), we propose sufficient conditions for explicitly constructing ranking functions and a distributed controller enforcing the satisfaction of a persistency specification by the abstracted interconnected system. Local distributed sensor information from other sub-systems is used for controller synthesis for each sub-system. Such partial information is characterized in terms of Lyapunov-like ranking functions. We provide also an algorithmic implementation of the whole process which will be applied to a numerical example of a distributed control of a three-tank system.

This paper is structured as follows. Some preliminary definitions are first given in Section 2. Section 3 is devoted to the problem statement. In Section 4, a procedure based on Lyapunov-like (or ranking) functions for building a reduced discrete abstraction of the original interconnected system is proposed. In Section 5, a procedure is proposed to compute those ranking functions and ensure that the persistency specification is satisfied. A step-by-step algorithm is provided. Section 6 presents a numerical example and some concluding remarks are given in Section 7.

2. Symbolic models and equivalence notions

2.1. Notation. The cardinality of a set A is denoted by $|A|$. The relative complement of the set A in the set B is denoted by $B \setminus A$. Given a relation $R \subseteq A \times B$ and $A_0 \subseteq A$, we define $R(A_0) = \{b \in B \,|\, \exists\, a \in A_0, (a,b) \in R\}$. $f : A \to B$ denotes an ordinary map, and $f^{-1}(b) := \{a \in A : f(a) = b\}$ for $b \in B$. The symbols $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{Z}$, $\mathbb{N}_0$ denote the set of real numbers, positive real numbers, integers, natural numbers including the zero, respectively. $[a;b] \subset \mathbb{Z}$ denotes a discrete interval with a and b as lower and upper bound, respectively. Given numbers $i, n \in \mathbb{N}$ with $i \leq n$, the following sets are defined where the element positions are given in ascending order with respect to their values: $\mathcal{I} = [1;n] \subset \mathbb{N}$, $\tilde{\mathcal{I}}^i = \mathcal{I} \setminus \{i\}$. Given two sets A and B, the product $A \times B$ denotes the Cartesian product, and for a collection of sets $\{A_j\}_{j \in \tilde{\mathcal{I}}^i}$, indexed by the set $\tilde{\mathcal{I}}^i$, the product $\prod_{j \in \tilde{\mathcal{I}}^i} A_j$ denotes the Cartesian product keeping the order in $\tilde{\mathcal{I}}^i$, i.e., $\prod_{j \in \tilde{\mathcal{I}}^i} A_j := A_1 \times \cdots \times A_{i-1} \times A_{i+1} \times \cdots \times A_n$. Given a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, x_i denotes the i-th component of x, and $\tilde{x}_i$ is defined as $\tilde{x}_i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

2.2. Transition systems. In this paper we will use the class of alternating transition systems as abstract models of control systems (Tabuada, 2009).

Definition 1. A *transition system* S is the quadruple (X_S, U_S, F_S, Y, H), where X_S is a set of states, U_S is a set of control inputs, $F_S \subseteq X_S \times U_S \times X_S$ is a *transition relation*, Y is a set of outputs, and $H : X_S \to Y$ is an output map. When $Y = X_S$ and the output map H is the identity function, then the transition system is reduced to the triple $S = (X_S, U_S, F_S)$.

F_S has the interpretation that a transition can occur from state x to state x' upon control input u if and only if $(x, u, x') \in F_S$. A transition $(x, u, x') \in F_S$ is also denoted by $x \xrightarrow{u} x'$. $Post_u(x) = \{x' \in X_S : (x, u, x') \in F_S\}$ denotes the set of successors of x upon control u.

Definition 2. Given two transition systems $S_a = (X_a, U_a, F_a)$ and $S_b = (X_b, U_b, F_b)$, a relation $R \subseteq X_a \times X_b$ is an *alternating simulation relation* from S_a to S_b if the following conditions are satisfied:

- $\forall\, x_b \in X_b, \exists\, x_a \in X_b, (x_a, x_b) \in R$,
- $(x_a, x_b) \in R$, $\forall u_a \in U_a$, $\exists u_b \in U_b, \forall x'_b \in Post_{u_b}(x_b), \exists x'_a \in Post_{u_a}(x_a)$, $(x'_a, x'_b) \in R$.

An alternating simulation relation allows the designer to work with the abstract system S_a instead of the concrete system S_b. For example, in the case of a reach-and-stay specification, if there is a suitable controller in S_a then there is one in S_b. For this, we need to assume that every state in S_b has a successor: $Post_{u_b}(x_b)$ is not empty for all x_b. Note also that the abstraction should not be too coarse: the states to reach-and-stay, as given by a specification, should be separated from other states in the abstraction. Under this condition, a controller for the abstract system S_a can be refined to a controller on the concrete system S_b; see Tabuada (2009).

Note that the feedback refinement relation as in the work of Reissig *et al.* (2017) is a special case of alternating simulation relations, where the designed controllers require quantized (or symbolic) state information only and can be interfaced with the initial system via a static quantizer. Contrarily to the approximate alternating simulation which is a one-sided relationship, approximate (alternating) bisimulation is a symmetric notion that bridges established notions in computer science and control theory (Girard and Pappas, 2011; Saoud, 2019).

3. Problem statement

Suppose that we are given an engineered system in the physical world (concrete system) characterized by a set of continuous-time differential equations:

$$\dot{\xi}(t) = g(\xi(t), \upsilon(t)) + w(t), \tag{1}$$

with state variable $\xi \in \mathcal{X} \subset \mathbb{R}^n$, control input $\upsilon \in \mathcal{U} \subset \mathbb{R}^m$, disturbance $w \in \llbracket W_{\min}, W_{\max} \rrbracket$ which denotes a hyper-interval $[W_{\min,1}, W_{\max,1}] \times \cdots \times [W_{\min,n}, W_{\max,n}]$ with $W_{\min} = [W_{\min,1}, \dots, W_{\min,n}]^T, W_{\max} = [W_{\max,1}, \dots, W_{\max,n}]^T \in \mathbb{R}^n$ and $\forall i \in \mathcal{I}, W_{\min,i} \leq W_{\max,i}$.

Assume that a discrete finite state model can be abstracted from the time-sampled version of the concrete system (1), and the abstraction is constructed by fixing a parameter τ for the sample time and vectors of parameters η, μ for the state and input spaces, using a feedback refinement relation between the concrete system and the abstract system (Reissig *et al.*, 2017). Notice that increasing τ, η and μ results in a smaller symbolic model (in terms of the cardinality of the state space and the number of elementary transitions between the symbolic states), but this could reduce the possibility of designing the desired control. The discrete non-deterministic system (2) is considered:

$$\forall\, i = 1, \dots, n : \; x_i^+ \in f_i(x_i, x_1, x_2, \dots, x_n, u_i) \tag{2}$$

with $x_i \in X_i$, $u_i \in U_i$ for some finite sets X_i, U_i, $i \in \mathcal{I}$. The above structure results from the interconnection of n subsystems. The system (2) is non-deterministic in the sense that if an input is applied in a state, several next states are possible. The trajectories of the system (2) are denoted by $\forall i \in \mathcal{I}$, $x_i(k, x_0, u_i)$ with initial condition $x_i(0, x_0, u_i) = x_0$, discrete time $k \in \mathbb{N}_0$, and control $c_i : X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} X_j \to U_i$ where the domain of the controller is defined as $dom(c) = \{x \in X \,|\, c(x) \neq \emptyset\}$.

Consider specifications of the form "reach P" or "reach P and stay there", where $P = \prod_{i \in \mathcal{I}} P_i$ for some sets $P_i \subseteq X_i$, using linear temporal logic notations these are written as $\Diamond P$ and $\Diamond\Box P$, respectively. It is *a priori* desired to find controllers $c_i : X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} X_j \to U_i$ for each $i \in \mathcal{I}$, such that the system described by (2), under the state feedback controls $u_i = c_i(x_i, \tilde{x}_i)$, satisfies

$$\begin{aligned}
&\Diamond P : \forall x_0 \in dom(c), \, \exists\, k_i \in \mathbb{N}_0, \;\; x_i(k_i, x_0, u_i) \in P_i, \\
&\Diamond\Box P : \forall x_0 \in dom(c), \exists\, k_i \in \mathbb{N}_0, \forall k \geq k_i, \\
&\qquad\qquad\qquad\qquad\qquad x_i(k, x_0, u_i) \in P_i.
\end{aligned}$$

The objective is to design a controller c_i for the subsystem i in a domain $X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} D_j$ of smaller cardinality than $X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} X_j$ by using *reduced knowledge* about other subsystems. Our approach is based on ranking functions that characterize partial information about the sensed states of other subsystems. The alternating simulation relation will be used to infer the existence of a controller for (2) from the existence of a controller for a reduced discrete abstraction.

4. Construction of reduced discrete abstractions

The transition system modeling the system (2) is denoted $S = (X_S, U_S, F_S)$ where $X_S = \prod_{i \in \mathcal{I}} X_i$, $U_S = \prod_{i \in \mathcal{I}} U_i$, and F_S is given by

$$F_S = \left\{ \begin{array}{c} (x, u, x') \in X_S \times U_S \times X_S : \\ \forall\, i \in \mathcal{I}, \; x_i' \in f_i(x_i, \tilde{x}_i, u_i) \end{array} \right\}. \tag{3}$$

The construction of a *reduced discrete abstraction* T based on ranking functions is done as follows: consider ranking functions defined by

$$V_i : X_i \to D_i \tag{4}$$

for each $i \in \mathcal{I}$, where $D_i = \{0, 1, 2, \dots, d_i\}$, for some $d_i \in \mathbb{N}_0$, and $|D_i| \leq |X_i|$. Intuitively, the ranking functions (4) represent some notion of distance to the set $V_i^{-1}(0) \subseteq X_i$, $i \in \mathcal{I}$. In principle, they can be freely proposed, but we will give a constructive way to build them when considering persistency specifications in the next section.

To begin with, the functions f_i of (2) defined in the domain $X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} X_j \times U_i$ are used to define the functions F_i over a simplified domain, as follows: $\forall i \in \mathcal{I}, \forall (x_i, \tilde{v}_i, u_i) \in X_i \times \prod_{j \in \tilde{\mathcal{I}}^i} D_j \times U_i$,

$$F_i(x_i, \tilde{v}_i, u_i) = \bigcup_{\tilde{x}_i \in \prod_{j \in \tilde{\mathcal{I}}^i} V_j^{-1}(v_j)} f_i(x_i, \tilde{x}_i, u_i). \tag{5}$$

Note that the relationship between states $\tilde{x}_i \in \prod_{j \in \tilde{\mathcal{I}}^i} X_j$ and values $\tilde{v}_i \in \prod_{j \in \tilde{\mathcal{I}}^i} D_j$ in (5) is determined by the ranking functions defined in (4). The controllable predecessor based on the values of ranking functions for each subsystems $i \in \mathcal{I}$ from a set $S_i \subseteq X_i$ under the influence of $\tilde{v}_i \in \prod_{j \in \tilde{\mathcal{I}}^i} D_j$ is defined as

$$CPre_i^{\tilde{v}_i}(S_i | U_i) = \left\{ \begin{array}{c} x_i \in X_i : \exists\, u_i \in U_i, \\ F_i(x_i, \tilde{v}_i, u_i) \subseteq S_i \end{array} \right\}. \tag{6}$$

The value $\tilde{v}_i$ in (6), which is defined from ranking functions (4), characterizes some partial information about the states of components other than i.

To simplify the notation, $V_i^{-1}(\leq v_i)$ will denote a shorthand for $\bigcup_{k \leq v_i} V_i^{-1}(k)$ with $k \in \mathbb{N}_0$. Consider the function $\mathbf{V}_i^{+}(v_i, \tilde{v}_i)$ for all $i \in \mathcal{I}$ defined with the controllable predecessor:

$$\mathbf{V}_i^{+}(v_i,\tilde{v}_i) = \min\left\{ \begin{array}{c} k \in \mathbb{N}_0 : V_i^{-1}(v_i) \subseteq \\ CPre_i^{\tilde{v}_i}\left(V_i^{-1}(\le k) \middle| U_i \right) \end{array} \right\}. \quad (7)$$

For such $\mathbf{V}_i^{+}$, consider the abstraction T given by $T = (X_T, U_T, F_T)$, where $X_T = \prod_{i\in\mathcal{I}} D_i$, $U_T = \{u_T\}$ and u_T is the only control input, and

$$F_T = \left\{ \begin{array}{c} (v, u_T, v') \in X_T \times \{u_T\} \times X_T : \\ \forall\, i \in \mathcal{I}, \;\; v_i' \le \mathbf{V}_i^{+}(v_i, \tilde{v}_i) \end{array} \right\}. \quad (8)$$

Lemma 1. (Apaza-Perez *et al.*, 2019) *The relation $R \subseteq X_T \times X_S$ given by*

$$R = \left\{ \;(v,x) \in X_T \times X_S : \forall\, i \in \mathcal{I},\; v_i = V_i(x_i)\; \right\}, \quad (9)$$

is an alternating simulation relation from T to S.

The next result gives the domain of admissible controllers for T satisfying the persistency specification.

Theorem 1. (Apaza-Perez *et al.*, 2019) *Suppose that T satisfies the specification $\Diamond\Box P_T$, for some set $P_T \subseteq X_T$. Then there exists a controller $c = (c_1, \dots, c_n)$, where c_i has domain $X_i \times \prod_{j\in\tilde{\mathcal{I}}^i} D_j$ enforcing the specification $\Diamond\Box P_S$ given by*

$$P_S = \{x \in X : (V_1(x_1), V_2(x_2), \dots, V_n(x_n)) \in P_T\}. \quad (10)$$

In this case, the controller can be chosen as follows:

$$\begin{aligned} & c_i(x_i, \tilde{v}_i) \\ & \quad \in \Big\{ u_i \in U_i : \max_{\tilde{x}_i \in \prod_{j\in\tilde{\mathcal{I}}^i} V_j^{-1}(v_j)} V_i(f_i(x_i, \tilde{x}_i, u_i)) \\ & \qquad\qquad \le \mathbf{V}_i^{+}(V_i(x_i), \tilde{v}_i) \Big\} \quad (11) \end{aligned}$$

Note that Theorem 1 is valid also for specifications $\Diamond P_T$, $\Diamond P_S$ instead of $\Diamond\Box P_T$, $\Diamond\Box P_S$.

Theorem 1 gives an explicit admissible set of controllers only when the system T satisfies the desired specification. Later, this property can be checked by analyzing cycles in T.

Definition 3. A *directed graph* $G = (\mathcal{V}, \mathcal{E})$ consists of a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a collection of ordered pairs. A *cycle* is a sequence of vertices $c(1), c(2), \dots, c(m) \in \mathcal{V}$ such that $c(1) = c(m)$ and $(c(i), c(i+1)) \in \mathcal{E}$ for all i. A cycle is called a *self-cycle* when $m = 2$, e.g., $(c(1), c(1)) \in \mathcal{E}$.

The system T defined by (7) and (8) can be considered as a directed graph $G = (X_T, \mathcal{E}_T)$, where $(v, v') \in \mathcal{E}_T$ if and only if $(v, u_T, v') \in F_T$. A necessary and sufficient condition under which system T enforces the specification $\Diamond\Box P_T$, for some set $P_T \subseteq X_T$, can be obtained in terms of cycle properties:

Proposition 1. *Consider T defined in* (7)–(8)*, a target set $P_T \subseteq X_T$. T satisfies the specification $\Diamond\Box P_T$ if and only if all cycles in $G = (X_T, \mathcal{E}_T)$ are included in P_T.*

Proof.
(Necessity) Assume that T satisfies $\Diamond\Box P_T$ and consider a cycle $v(1), v(2), \dots, v(m) = v(1)$ that is reachable from some initial state $w(1)$, through a path $w(1) \to w(2) \to \cdots \to w(k) = v(1)$. Since the infinite path $w(1) \to w(2) \to \cdots \to v(1) \to \cdots \to v(m-1) \to \dots$ reaches P_T and stays there forever, all vertices of the cycle must belong to P_T.

(Sufficiency) This is proved by contradiction: assume that all reachable cycles are included in P_T, and consider an infinite path from some initial state. If this path does not satisfy $\Diamond\Box P_T$ then some vertex that appears infinitely often on this path does not belong to P_T. But this means that some reachable cycle is not included in P_T, a contradiction. ■

The following proposition gives a sufficient condition under which the ranking functions give a system T satisfying the specification reach-and-stay $\Diamond\Box P_T$, for some set $P_T = P_{1T} \times \cdots \times P_{nT} \subseteq X_T$. The stay part of $\Diamond\Box P_T$ is guaranteed by condition (i) and the reach part by (ii), (iii).

Proposition 2. *Consider the system $T = (X_T, U_T, F_T)$ defined from* (7) *and* (8) *and any downward closed set $P_T \subseteq X_T$, i.e., for every $(v(1), \dots, v(n)) \in P_T$, if $w(i) \le v(i)$ for all $i \in \mathcal{I}$, then $(w(1), \dots, w(n)) \in P_T$.*
T satisfies the specification $\Diamond\Box P_T$ if the following conditions are satisfied:

(i) $\forall v \in P_T$, $\left(\mathbf{V}_1^{+}(v_1, \tilde{v}_1), \dots, \mathbf{V}_n^{+}(v_n, \tilde{v}_n)\right) \in P_T$,

(ii) $\forall\, v \in X_T \setminus P_T$, $\forall i \in \mathcal{I}$, $\mathbf{V}_i^{+}(v_i, \tilde{v}_i) \le v_i$,

(iii) $\forall v \in X_T \setminus P_T$, $\exists i \in \mathcal{I}$, $\mathbf{V}_i^{+}(v_i, \tilde{v}_i) < v_i$.

Proof. The proof of Proposition 2 must ensure that all cycles consist of vertices in P_T according to Proposition 1. Condition (i) implies that there are no cycles with vertices in P_T and $X_T \setminus P_T$ simultaneously. The argument is reduced to two claims. The first one shows that there are no cycles with more than one vertex in $X_T \setminus P_T$, and the second one shows that there are no self-cycles in $X_T \setminus P_T$.

Claim 1. There are no cycles (except maybe self-cycles) in T with vertices in $X_T \setminus P_T$.

Consider a cycle

$$v(1) \xrightarrow{u_T} v(2) \xrightarrow{u_T} v(3) \quad (12)$$

in T outside of P_T, where $v(1) = v(3)$.

The relation in (8) implies $v_i(2) \le \mathbf{V}_i^{+}(v_i(1), \tilde{v}_i(1))$ and $v_i(3) \le \mathbf{V}_i^{+}(v_i(2), \tilde{v}_i(2))$. From the condition (i) and considering the cycle condition $v(1) = v(3)$, one can conclude $v(1) = v(2)$. Analogously this procedure can be extended to $v(1) \xrightarrow{u_T} v(2) \xrightarrow{u_T} \dots \xrightarrow{u_T} v(m)$ for arbitrary values of m, proving $v(1) = v(2) = \cdots = v(m)$.

Claim 2. There are no self-cycles in T with vertices in $X_T \setminus P_T$.

This claim is proved by contradiction: let $v \in X_T \setminus P_T$ and suppose that there is a self-cycle in v. This means $(v, \{u_T\}, v) \in F_T$. Using the relation (8), it is equivalent to $v_i \leq \mathbf{V}_i^+(v_i, \tilde{v}_i)$ for all $i \in \mathcal{I}$, this is a contradiction with the condition (iii).

These claims guarantee that all cycles are included in P_T. Thus, T satisfies the specification $\Diamond\Box P_T$ by Proposition 1. ■

This section has shown that it is possible to build reduced discrete abstractions from the ranking functions and these can provide an explicit admissible set of controllers only when the specifications are satisfied. The next section will address the problem of constructing ranking functions ensuring that the reduced system satisfies the reach, reach and stay specifications.

5. Ranking functions and persistency specifications

The definition of predecessor (6) relies on the ranking functions $\{V_i\}_{i\in\mathcal{I}}$ and we introduce a generic definition for building ranking functions. The controllable predecessors $CP_i\,(U_i, E, S_i)$, where U_i is the input set, $E \subseteq \prod_{j\in\tilde{\mathcal{I}}^i} X_j$ and $S_i \subseteq X_i$, is defined by

$$CP_i\,(U_i, E, S_i) = \left\{ \begin{array}{c} x_i \in X_i : \exists\, u_i \in U_i, \forall \tilde{x}_i \in E, \\ f_i(x_i, \tilde{x}_i, u_i) \subseteq S_i \end{array} \right\}. \tag{13}$$

The controllable predecessor defined in (13) describes the states in X_i for which the controlled system i is able to reach the target set S_i despite the influences (expressed by E) of other interconnected systems (robustness property). E may depend on some available partial knowledge about states of other components.

The predecessor in (13) does not depend on ranking functions defined *a priori*. When ranking functions $\{V_i\}_{i\in\mathcal{I}}$ are given, and $v \in \prod_{i\in\mathcal{I}} D_i$ (the domain of ranking functions), then taking $E = \prod_{j\in\tilde{\mathcal{I}}^i} V_j^{-1}(v_j)$ allows us to recover the predecessor defined in (6), through

$$CP_i\left(U_i, \prod_{j\in\tilde{\mathcal{I}}^i} V_j^{-1}(v_j), S_i\right) = CPre_i^{\tilde{v}_i}\,(S_i|U_i)\,. \tag{14}$$

The controllable predecessor given in (13) allows us to find ranking functions based on Algorithm 1 which satisfy the specifications (Theorem 2). The algorithm uses the controllable predecessor defined in (13) and some properties are required to define the ranking functions, which are obtained from the following lemma.

Lemma 2. *The controllable predecessor satisfies*

(i) $E_{i_1} \subseteq E_{i_2} \Rightarrow CP_i\,(U_i, E_{i_2}, S_i) \subseteq CP_i\,(U_i, E_{i_1}, S_i)$

(ii) $S_{i_1} \subseteq S_{i_2} \Rightarrow CP_i\,(U_i, E_i, S_{i_1}) \subseteq CP_i\,(U_i, E_i, S_{i_2})$.

Proof.
(i) Let $x_i \in CP_i\,(U_i, E_{i_2}, S_i)$. Then $\exists\, u_i \in U_i, \forall \tilde{x}_i \in E_{i_2}, f_i(x_i, \tilde{x}_i, u_i) \subseteq S_i$. Due to the condition $E_{i_1} \subseteq E_{i_2}$, one can ensure that

$$\exists\, u_i \in U_i, \forall \tilde{x}_i \in E_{i_1}, f_i(x_i, \tilde{x}_i, u_i) \subseteq S_i,$$

which implies $x_i \in CP_i\,(U_i, E_{i_1}, S_i)$ by (13).

(ii) Let $x_i \in CP_i\,(U_i, E_i, S_{i_1})$. Then $\exists\, u_i \in U_i, \forall \tilde{x}_i \in E_i, f_i(x_i, \tilde{x}_i, u_i) \subseteq S_{i_1}$. Since $S_{i_1} \subseteq S_{i_2}$, one can ensure that

$$\exists\, u_i \in U_i, \forall \tilde{x}_i \in E_i, f_i(x_i, \tilde{x}_i, u_i) \subseteq S_{i_2},$$

which implies $x_i \in CP_i\,(U_i, E_i, S_{i_2})$ by (13). ■

Notation. We assume $\mathbf{b}(c) = 1$ if c is true, $\mathbf{b}(c) = 0$ otherwise; $Z_i(\leq k) = \bigcup_{l\leq k} Z_i(l)$, $\{< i\} = \{\hat{j} \in \mathcal{I} : \hat{j} < i\}$, and $\{\leq i\}, \{> i\}$ are defined analogously. The value ℓ_s in (17) and (18) denotes the s-th entry of vector $\ell \in L(i) \subseteq \mathbb{R}^n$.

The idea of Algorithm 1 is inspired by Lyapunov functions and level sets from classical control theory. Intuitively, each set $Z_i(k)$ can be considered as a level set. In each iteration k in Step 3, the search for new level sets $Z_i(k)$ is based on two principles: (i) given the system i, the set $Z_i(k)$ in (17) reaches lower levels, which is denoted $Z_i(\leq k-1)$, despite the influence of the sets $Z_s(\ell_s)$ on the other systems (which are already defined by previous iterations); (ii) the condition in (18) shows that the set $Z_i(k)$ will be considered in our sequence when its effect on the other sets Z_j does not cause any increase in levels. This idea is illustrated in Fig. 1, where T_i denotes the set $Z_i(k)$, and the arrows and circles describe possible behaviors in the level sets: the arrow indicates that the set reaches strictly lower levels (decreasing behavior), the arrow with a circle indicates that it is possible to reach lower levels or stay in the same level (non-increasing behavior).

Regarding Step 2 in Algorithm 1, the computation of feasible contracts for invariance (or safety) can be used to find the sets B_i; e.g., Zonetti *et al.* (2019) propose

Algorithm 1. Building a sequence of sets $Z_i(k) \subseteq X_i$, $i \in \mathcal{J}$, $k \in \mathbb{N}_0$ and controller domain.

Step 1. Initialization with the target sets related for each subsystem:

$$Z_1(0) \leftarrow P_1, \dots, Z_i(0) \leftarrow P_i, \dots, Z_n(0) \leftarrow P_n; \quad (15)$$

Step 2. Find sets $B_i \subseteq P_i, \forall\, i \in \mathcal{J}$ such that

$$B_i \subseteq CP_i\left(U_i, \prod_{s \in \tilde{\mathcal{J}}^i} Z_s(0), B_i\right). \qquad (16)$$

Step 3. Find a sequence of sets in each subsystem, which reach the target sets despite the influence of other subsystems.

```
$k \leftarrow 0$;   for $i \leftarrow 1, n$ do $Z_i(0) \leftarrow B_i$ end for;
while $\exists i \in \mathcal{J}, Z_i(k) \neq \emptyset$ do
    $k \leftarrow k + 1$;
    for $i \leftarrow 1, n$ do
        $L(i) \leftarrow \{0, 1, ..., k\}^{(i-1)} \times \{k\} \times \{0, 1, ..., k-1\}^{(n-i)}$;
        $Z_i(k) \leftarrow \bigcap_{\ell \in L(i)} CP_i\left(U_i, \prod_{s \in \tilde{\mathcal{J}}^i} Z_s(\ell_s), Z_i(\leq k-1)\right)$;   (17)
        cond = $\forall \ell \in L(i),\ \forall\, j \in \tilde{\mathcal{J}}^i, \forall q \in \{\leq k - \mathbf{b}\,(j > i)\}$,
            $Z_j(q) \subseteq CP_j\left(U_j, \prod_{s \in \tilde{\mathcal{J}}^j} Z_s(\ell_s), Z_j(\leq q)\right)$;   (18)
        If cond $\wedge$ not $(Z_i \subseteq Z_i(\leq k-1))$, then
            $Z_i(k) \leftarrow Z_i(k) \setminus Z_i(\leq k-1)$;   (19)
        else
            $Z_i(k) \leftarrow \emptyset$;
        end if
    end for
end while
```

an approach based on the monotonicity property; Eqtami and Girard (2019) base their the approach on quantitative computation of controlled invariants; in the works of Ghasemi *et al.* (2020) the approach is based on convexity properties; Chen *et al.* (2019) propose an approach based on an epigraph method. (16) ensures that the specification "stay in P_T" is fulfilled (see also Theorem 2).

Remark 1. The domain of the distributed controller $dom(c)$ resulting from Algorithm 1 can be explicitly characterized as

$$dom(c) = \prod_{i \in \mathcal{J}} Z_i(\leq k_{\max}(i)), \qquad (20)$$

where $\forall i \in \mathcal{J}, k_{\max}(i) := \max\{k \in \mathbb{N}_0 | Z_i(k) \neq \emptyset\}$. Note that, $\emptyset$ denoting the empty set (to be distinguished from a non-empty "zero" value) and provided $f_i(x_i, \tilde{x}_i, u_i)$ satisfies $\forall (x_i, u_i), f_i(x_i, \emptyset, u_i) = \emptyset$, then $CP_i(U_i, \emptyset, S_i) = X_i$ from (13).

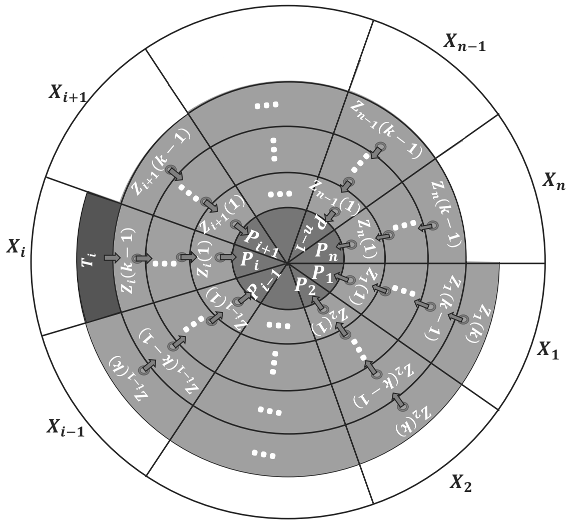

Fig. 1. Algorithm 1: k-th iteration for the system i. T_i denotes the set $Z_i(k)$ in (17), the arrows and circles describe possible behaviors in the sets of Z_j's (level sets): the arrow indicates that the set T_i reaches lower levels, the arrow with a circle indicates that other Z_j's reach lower levels or stay on the same level.

Since the X_i's are finite sets, the completion of the procedure in Algorithm 1 can be characterized by the following: $\forall i \in \mathcal{J}$, $\exists\, l \in \mathbb{N}$, $k \geq l$, $Z_i(k) = \emptyset$. Consequently, we define

$$d_i := \min\{k \in \mathbb{N}_0 : Z_i(k+1) = \emptyset\}. \qquad (21)$$

Let $X_i^Z := \cup_{k \in [0; d_i]} Z_i(k)$ be the domain where the ranking functions are defined, and $X^Z = \prod_{i \in \mathcal{J}} X_i^Z$. $X_T^Z = \prod_{i \in \mathcal{J}} \{0, 1, \dots, d_i\}$ denotes the state space used to build the reduced abstraction according to Section 4.

Lemma 3. *Consider the sets Z_i obtained from Algorithm 1. The ranking functions V_i defined in X_i^Z as*

$$\forall\, i \in \mathcal{J}, \quad V_i(x) = \min\{k \in \mathbb{N}_0 : x \in Z_i(k)\}, \qquad (22)$$

satisfy

$$\forall v_i \in [0, \dots, d_i], \quad V_i^{-1}(v_i) = Z_i(v_i). \qquad (23)$$

The proof of Lemma 3 is immediately obtained by condition (19) in Algorithm 1, which ensures a property of disjoint sets in the sets Z_i.

Theorem 2. *Consider the system* (2) *and the sets Z_i according to Algorithm 1. Define the ranking functions $V_i : X_i^Z \subseteq X_i \to \mathbb{N}_0$, $i \in \mathcal{J}$ based on the sets Z_i as in* (22) *and define a system T as in* (7) *and* (8). *Let*

$X^Z = \prod_{i\in\Im} X_i^Z$. Then, T satisfies the specification $\Diamond P_T$ in X^Z, and the truth of (16) *also ensures that T satisfies $\Diamond\Box P_T$.*

Proof. It is based on Proposition 2, where the three conditions are expressed by the following three claims.

Claim 1. If the "stay" condition (16) in Algorithm 1 is true, then

$$\forall v \in P_T, \left(\mathbf{V}_1^+(v_1,\tilde{v}_1),\dots,\mathbf{V}_n^+(v_n,\tilde{v}_n)\right) \in P_T.$$

According to (15) and (22) the ranking functions satisfy

$$\forall\, i \in \Im,\ \ V_i^{-1}(0) = P_i, \tag{24}$$

which implies $P_T = \{\mathbf{0}\}$ where $\mathbf{0} \in X_T$ is the zero vector. In terms of ranking functions, Step 2 in Algorithm 1 implies

$$\forall\, i \in \Im,\ V_i^{-1}(0) \subseteq CP_i\left(U_i, \textstyle\prod_{s\in\tilde{\Im}^i} V_s^{-1}(0), V_i^{-1}(0)\right). \tag{25}$$

Note that $CP_i\left(U_i, \prod_{s\in\tilde{\Im}^i} V_s^{-1}(0), V_i^{-1}(0)\right) = CPre_i^{\tilde{\mathbf{0}}_i}\left(V_i^{-1}(0)|U_i\right)$ is satisfied by (14), from which the inclusion (25) can be expressed as

$$\forall\, i \in \Im,\ V_i^{-1}(0) \subseteq CPre_i^{\tilde{\mathbf{0}}_i}\left(V_i^{-1}(0)|U_i\right). \tag{26}$$

The inclusion (26) ensures (27) according to (7):

$$\forall\, i \in \Im,\ \ \mathbf{V}_i^+(\mathbf{0}) = 0. \tag{27}$$

Claim 2. $\forall\, v \in X_T^Z \setminus P_T, \forall i \in \Im, \mathbf{V}_i^+(v_i,\tilde{v}_i) \le v_i$.

Let $v \in X_T^Z \setminus P_T$, then $\forall\, i \in \Im$, $v_i \in [0; d_i]$. Define

$$\begin{aligned} \hat{v}_m &= \max\{v_i : i \in \Im\},\\ r &= \max\{i \in \Im : v_i = \hat{v}_m\}, \end{aligned} \tag{28}$$

which implies that $v_i \le \hat{v}_m$ for $i \le r$, and $v_i < \hat{v}_m$ for $i > r$.

Note that $v_r = \hat{v}_m > 0$, and consider the k-th iteration with $k = \hat{v}_m$ and the subsystem r in Algorithm 1, giving $T_r = Z_r(v_r)$,

$$T_r = \bigcap_{\ell\in L(r)} CP_r(U_r, \prod_{s\in\tilde{\Im}^r} Z_s(\ell_s), Z_r(\le v_r - 1)), \tag{29}$$

$$\begin{aligned} L(r) = \{0,1,\dots,\tilde{v}_m\}^{(r-1)} &\times \{\tilde{v}_m\}\\ &\times \{0,1,\dots,\tilde{v}_m - 1\}^{(n-r)}, \end{aligned}$$

$$Z_j(v_j) \subseteq CP_j\left(U_j, \prod_{s\in\tilde{\Im}^j} Z_s(v_s), Z_j(\le v_j)\right), \tag{30}$$

$\forall j \in \tilde{\Im}^r$, by (17) and (18).

Lemma 3 implies $V_i^{-1}(\le\ v_i) = Z_i(\le\ v_i)$ and $\prod_{s\in\tilde{\Im}^j} V_s^{-1}(v_s) = \prod_{s\in\tilde{\Im}^j} Z(v_s)$, ensuring

$$\begin{aligned} \forall j \in \tilde{\Im}^r,\ \ CP_j&\left(U_j, \prod_{s\in\tilde{\Im}^j} Z_s(v_s), Z_j(\le v_j)\right)\\ &= CP_j\left(U_j, \prod_{s\in\tilde{\Im}^j} V_s^{-1}(v_s), V_j^{-1}(\le v_j)\right). \end{aligned} \tag{31}$$

For each $j \in \tilde{\Im}^r$, we have that $V_j^{-1}(v_j) = Z_j(v_j)$ and $CP_j\left(U_j, \prod_{s\in\tilde{\Im}^j} V_s^{-1}(v_s), V_j^{-1}(\le v_j)\right) = CPre_j^{\tilde{v}_j}\left(V_j^{-1}\ (\le v_j)|\,U_j\right)$ are obtained by Lemma 3 and (14) respectively, which imply

$$\forall j \in \tilde{\Im}^r,\ \ V_j^{-1}(v_j) \subseteq CPre_j^{\tilde{v}_j}\left(V_j^{-1}\ (\le v_j)|\,U_j\right), \tag{32}$$

from (30) and (31). Consequently, $\mathbf{V}_j^+(v_j,\tilde{v}_j) \le v_j, \forall j \in \tilde{\Im}^r$ is satisfied from (7).

Moreover, the inclusion

$$\begin{aligned} \bigcap_{\ell\in L(r)} CP_r&\left(U_r, \prod_{s\in\tilde{\Im}^r} Z_s(\ell_s), Z_r(\le v_r - 1)\right)\\ &\subseteq CP_r\left(U_r, \prod_{s\in\tilde{\Im}^r} Z_s(v_s), Z_r(\le v_r - 1)\right), \end{aligned} \tag{33}$$

is guaranteed by (28) due to the relations $v_i \le \hat{v}_m$ for $i \le r$ and $v_i < \hat{v}_m$ for $i > r$. Lemma 3 guarantees

$$\begin{aligned} CP_r&\left(U_r, \prod_{s\in\tilde{\Im}^r} Z_s(v_s), Z_r(\le v_r - 1)\right)\\ &\subseteq CP_r\left(U_r, \prod_{s\in\tilde{\Im}^r} V_s^{-1}(v_s), V_r^{-1}(< v_r)\right). \end{aligned} \tag{34}$$

The following relation is obtained from (14):

$$\begin{aligned} CP_r&\left(U_r, \prod_{s\in\tilde{\Im}^r} V_s^{-1}(v_s), V_r^{-1}(< v_r)\right) =\\ &= CPre_r^{\tilde{v}_r}\left(V_r^{-1}\ (< v_r)|\,U_r\right). \end{aligned} \tag{35}$$

Thus, from (29), (33), (34) and (35) we get

$$\mathbf{V}_r^+(v_r,\tilde{v}_r) < v_r. \tag{36}$$

Finally, the following condition is ensured by (32) and (36):

$$\forall\, v \in X_T^Z \setminus P_T, \forall i \in \Im, \mathbf{V}_i^+(v_i,\tilde{v}_i) \le v_i. \tag{37}$$

Claim 3. $\forall v \in X_T^Z \setminus P_T, \exists i \in \Im, \mathbf{V}_i^+(v_i,\tilde{v}_i) < v_i$.

Claim 3 is guaranteed from a process analogous to that of case $i = r$ in Claim 2, where (28), (29) (33) and (34) imply

$$\mathbf{V}_r^+(v_r, \tilde{v}_r) < v_r.$$

Claims 1–3 satisfy the conditions of Proposition 2, which imply that the system T, according to (7) and (8), enforces the specification $\Diamond\Box P_T$ in X^Z. ■

Consequently, Theorem 2 provides a constructive way to satisfy the main assumptions of Theorem 1, so giving an explicit procedure for controller synthesis.

6. Numerical example

In this section, the theoretical results of this paper are illustrated through distributed control of a three-tank system. Figure 2 shows the diagram of a coupled three-tank system. This example has been extensively used in the control literature as a benchmark (see, e.g., Zolghadri *et al.*, 1996). The system consists of three cylinders T_1, T_2 and T_3; these are connected by cylindrical pipes with a circular cross-section of area S_C and outflow coefficients of Tanks 1 and 2 are a_{z1} and a_{z2}, respectively. The nominal inflows (q_1 and q_2) are located at Tanks 1 and 3, respectively. The inflow rate can be continuously manipulated from 0 to a maximum flow rate of $q_{\max}$ to maintain the tank level. The measured variables are the level of Tank 1 (h_1), Tank 2 (h_2) and Tank 3 (h_3). The nominal outflow pipe has a cross section S_C with an outflow coefficient a_{z3} and located at Tank 3. The control objective is to control the levels of Tanks 1 and 3 by manipulating the inflow rates q_1 and q_2.

The three-tank system represented using the mass balance is given by

$$\begin{aligned}\dot{h}_1(t) &= \frac{1}{S}(q_1 - S_1 a_{z1}\mathrm{sign}(h_1 - h_2)\sqrt{2g(h_1 - h_2)}),\\ \dot{h}_2(t) &= \frac{1}{S}\Big(S_1 a_{z1}\mathrm{sign}(h_1 - h_2)\sqrt{2g(h_1 - h_2)}\\ &\quad - S_2 a_{z2}\mathrm{sign}(h_2 - h_3)\sqrt{2g(h_2 - h_3)}\Big),\\ \dot{h}_3(t) &= \frac{1}{S}\Big(q_2 - S_2 a_{z2}\mathrm{sign}(h_2 - h_3)\sqrt{2g(h_2 - h_3)}\\ &\quad - S_3 a_{z3}\sqrt{2ah_3}\Big). \qquad (38)\end{aligned}$$

The physical parameters of the three tank system are presented in Table 1.

The linearized state-space model in continuous form is

$$\dot{x}(t) = Ax(t) + Bu(t). \qquad (39)$$

The linearization technique is valid in the vicinity of an operating point. The nonlinear system (38) is linearized around the following steady state operating point: $\begin{bmatrix} h_{1o} & h_{2o} & h_{3o}\end{bmatrix} = \begin{bmatrix}0.6 & 0.5 & 0.4\end{bmatrix}^T$ m and $\begin{bmatrix}q_{1o} & q_{2o}\end{bmatrix}^T = \begin{bmatrix}0.35787 & 0.6563\end{bmatrix}^T \times 10^{-4}\,\mathrm{m}^3/\mathrm{s}$. The continuous state space model for the parameters in Table 1 is

$$A = \begin{bmatrix} -0.01046 & 0.01046 & 0\\ 0.01046 & -0.02127 & 0.01081\\ 0 & 0.01081 & -0.0183\end{bmatrix},$$

$$B = \begin{bmatrix} 58.4795 & 0\\ 0 & 0\\ 0 & 58.4795\end{bmatrix}. \qquad (40)$$

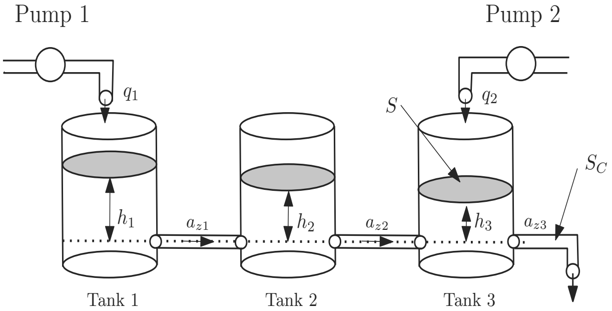

Fig. 2. Structure of the three-tank system.

Table 1. Physical parameters of the three-tank system.

Parameters	Values
Tank cross-section area	$S = 0.0171$ m^2
Pipe cross-section area	$S_C = S_1 = S_2 = S_3$ $= 0.00005$ m^2
Pipe outflow coefficients	$a_{z1} = 0.511$, $a_{z2} = 0.5279$ $a_{z3} = 0.7313$
Maximum level	$h_{\max} = 0.68$ m
Maximum in-flow rate	$q_{\max} = 1.2 \times 10^{-4}$ m^3/s

In the first step, the tool SCOTS (Rungger and Zamani, 2016) is used to abstract the infinite state system from (38) as a finite transition system. SCOTS constructs non-deterministic transition systems, which are obtained using a state-quantization parameter η, an input-quantization parameter μ, and a time-quantization parameter τ. Three admissible values for the control signals q_i's have been selected around the operating point: $\forall i \in \{1,2\}$, $U_i = \{\underline{q_{io}}, \frac{1}{2}(\underline{q_{io}} + \overline{q_{io}}), \overline{q_{io}}\}$ with $\underline{q_{1o}} = 0.1 \times 10^{-4}$, $\overline{q_{1o}} = 0.7 \times 10^{-4}$, $\underline{q_{2o}} = 0.3 \times 10^{-4}$, $\overline{q_{2o}} = 0.9 \times 10^{-4}$. The quantization parameters are considered as follows: $\forall i \in \{1,2\}, j \in \{1,2,3\}$, $\eta_j = 0.005$, $\mu_i = \frac{1}{2}(\overline{q_{io}} - \underline{q_{io}})$ and $\tau = 0.1$ s. The transition relation (3) can be expressed through a matrix F_S where columns represent the tank levels, controls and next tank levels, and the rows correspond to the transitions.

The control objective is to satisfy the specification to reach and stay in the target set defined as follows: $0.59 \le h_1 \le 0.61, 0.48 \le h_2 \le 0.52, 0.39 \le h_3 \le 0.41.$

Thus, Algorithm 1 is initialized at Step 1 (see (15)) with the discrete target sets given by $P_1 = \eta_1\mathbb{Z} \cap [0.59, 0.61]$, $P_2 = \eta_2\mathbb{Z} \cap [0.48, 0.52]$, $P_3 = \eta_3\mathbb{Z} \cap [0.39, 0.41]$. The computation of the sets $Z_i(k)$ from Algorithm 1 leads to the following maximum values for the co-domain of the ranking functions: $d_1 = 4$, $d_2 = 3$, $d_3 = 6$, see (4). The state space can be decomposed using level sets of ranking functions, see Fig. 3. As can be seen in Fig. 3, the level sets are represented around the target set (dark gray set). Next, the admissible values for each controller q_i with $i = 1, 2$ have been computed (see (11)). This results in a reduced domain $X_i \times \prod_{j\in\{1,2,3\}\setminus\{i\}} D_j$ which is illustrated in Fig. 4. This numerical example shows that the distributed symbolic control can be designed in a step-by-step way based on user-chosen design parameters. The controller can be implemented using a lazy evolution strategy (Mazo *et al.*, 2010).

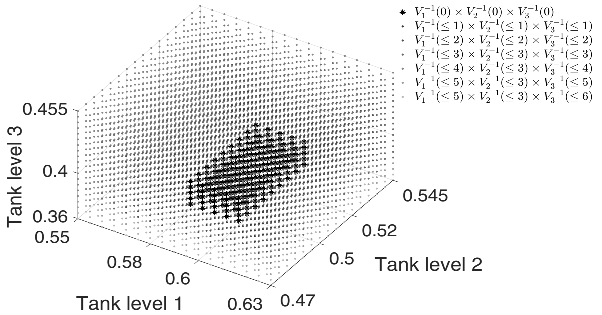

Fig. 3. Level sets corresponding to the ranking functions, where the dark gray set in the center is the target set.

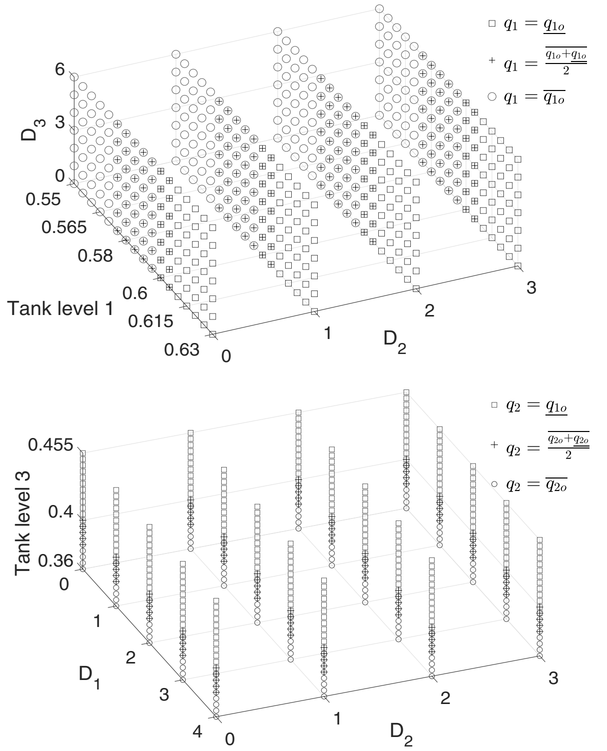

Fig. 4. Admissible control values in a reduced domain.

7. Conclusions

In this paper, the problem of distributed control design for interconnected systems is addressed. An advantage of the presented control synthesis methodology is that the knowledge of the full state is not required: the ranking functions provide partial information used in the abstraction considered. This results in lower complexity controllers for each sub-system. An algorithmic procedure was proposed to implement the method also illustrated through a numerical example. The optimization of the ranking functions calculation to improve the scalability of the proposed method is a topic of our current research.

Acknowledgment

This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the framework of the *Investments for the Future* Programme IdEx Bordeaux—SysNum (ANR-10-IDEX-03-02).

The authors would like to thank Dr. I. Walukiewicz and Prof. A. Muscholl (LaBRI, University of Bordeaux) for many insightful discussions on this work and their helpful comments.

References

Apaza-Perez, W.A., Combastel, C. and Zolghadri, A. (2019). Abstraction-based low complexity controller synthesis for interconnected non-deterministic systems, *18th European Control Conference (ECC), Naples, Italy*, pp. 4174–4179.

Belta, C., Yordanov, B. and Göl, E. (2017). *Formal Methods for Discrete-Time Dynamical Systems*, Springer, Cham.

Borri, A., Pola, G. and Benedetto, M.D.D. (2012). A symbolic approach to the design of nonlinear networked control systems, *15th ACM International Conference on Hybrid Systems: Computation and Control, HSCC'12, Beijing, China*, pp. 255–264.

Borri, A., Pola, G. and Benedetto, M.D.D. (2019). Design of symbolic controllers for networked control systems, *IEEE Transactions on Automatic Control* **64**(3): 1034–1046.

Chen, Y., Anderson, J., Kalsi, K., Low, S.H. and Ames, A.D. (2019). Compositional set invariance in network systems with assume-guarantee contracts, *2019 American Control Conference (ACC), Philadelphia, PA, USA*, pp. 1027–1034.

Coënt, A.L., Fribourg, L., Markey, N., Vuyst, F.D. and Chamoin, L. (2016). Distributed synthesis of state-dependent switching control, *in* K.G. Larsen *et al.* (Eds), *Reachability Problems*, Springer, Cham, pp. 119–133.

Dallal, E. and Tabuada, P. (2015). On compositional symbolic controller synthesis inspired by small-gain theorems, *54th IEEE Conference on Decision and Control (CDC), Osaka, Japan,* pp. 6133–6138.

Dashkovskiy, S.N., Rüffer, B.S. and Wirth, F.R. (2010). Small gain theorems for large scale systems and construction of ISS Lyapunov functions, *SIAM Journal on Control and Optimization* **48**(6): 4089–4118.

Eqtami, A. and Girard, A. (2019). A quantitative approach on assume-guarantee contracts for safety of interconnected systems, *18th European Control Conference (ECC), Naples, Italy*, pp. 536–541.

Ge, X., Yang, F. and Han, Q.L. (2017). Distributed networked control systems: A brief overview, *Information Sciences* **380**: 117–131.

Ghasemi, K., Sadraddini, S. and Belta, C. (2020). Compositional synthesis via a convex parameterization of assume-guarantee contracts, *23rd International Conference on Hybrid Systems: Computation and Control, HSCC'20, Sydney, Australia.*

Girard, A., Gössler, G. and Mouelhi, S. (2016). Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models, *IEEE Transactions on Automatic Control* **61**(6): 1537–1549.

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

Gruber, F., Kim, E.S. and Arcak, M. (2017). Sparsity-aware finite abstraction, *2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia*, pp. 2366–2371.

Henzinger, T.A., Qadeer, S., Rajamani, S.K. and Tasiran, S. (2002). An assume-guarantee rule for checking simulation, *ACM Transactions on Programming Languages and Systems* **24**(1): 51–64.

Jabri, D., Guelton, K., Belkhiat, D.E.C. and Manamanni, N. (2020). Decentralized static output tracking control of interconnected and disturbed Takagi–Sugeno systems, *International Journal of Applied Mathematics and Computer Science* **30**(2): 225–238, DOI: 10.34768/amcs-2020-0018.

Jiang, Z.P., Teel, A.R. and Praly, L. (1994). Small-gain theorem for ISS systems and applications, *Mathematics of Control, Signals and Systems* **7**(2): 95–120.

Kamiński, M. (2015). Symbolic computing in probabilistic and stochastic analysis, *International Journal of Applied Mathematics and Computer Science* **25**(4): 961–973, DOI: 10.1515/amcs-2015-0069.

Kim, E.S., Arcak, M. and Seshia, S.A. (2015). Compositional controller synthesis for vehicular traffic networks, *2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan*, pp. 6165–6171.

Majumdar, R. and Zamani, M. (2012). Approximately bisimilar symbolic models for digital control systems, *in* P. Madhusudan and S.A. Seshia (Eds), *Computer Aided Verification*, Springer, Berlin/Heidelberg, pp. 362–377.

Mazo, M., Davitian, A. and Tabuada, P. (2010). PESSOA: A tool for embedded controller synthesis, *in* T. Touili *et al.* (Eds), *Computer Aided Verification*, Springer, Berlin/Heidelberg, pp. 566–569.

Meyer, P. and Dimarogonas, D.V. (2017). Compositional abstraction refinement for control synthesis under lasso-shaped specifications, *2017 American Control Conference (ACC), Seattle, WA, USA*, pp. 523–528.

Meyer, P., Girard, A. and Witrant, E. (2018). Compositional abstraction and safety synthesis using overlapping symbolic models, *IEEE Transactions on Automatic Control* **63**(6): 1835–1841.

Mouelhi, S., Girard, A. and Gossler, G. (2013). COSYMA: A tool for controller synthesis using multi-scale abstractions, *16th International Conference on Hybrid Systems: Computation and Control, HSCC'13, Philadelphia, PA, USA*, pp. 83–88.

Nilsson, L.P. (2017). *Correct-by-Construction Control Synthesis for High-Dimensional Systems*, PhD thesis, University of Michigan, Ann Arbor, MI.

Nilsson, P. and Ozay, N. (2020). Control synthesis for permutation-symmetric high-dimensional systems with counting constraints, *IEEE Transactions on Automatic Control* **65**(2): 461–476.

Pola, G., Girard, A. and Tabuada, P. (2008). Approximately bisimilar symbolic models for nonlinear control systems, *Automatica* **44**(10): 2508–2516.

Pola, G., Pepe, P. and Benedetto, M.D.D. (2018). Decentralized supervisory control of networks of nonlinear control systems, *IEEE Transactions on Automatic Control* **63**(9): 2803–2817.

Pola, G., Pepe, P., Benedetto, M.D.D. and Tabuada, P. (2010). Symbolic models for nonlinear time-delay systems using approximate bisimulations, *Systems & Control Letters* **59**(6): 365–373.

Reissig, G. (2011). Computing abstractions of nonlinear systems, *IEEE Transactions on Automatic Control* **56**(11): 2583–2598.

Reissig, G., Weber, A. and Rungger, M. (2017). Feedback refinement relations for the synthesis of symbolic controllers, *IEEE Transactions on Automatic Control* **62**(4): 1781–1796.

Rungger, M. and Zamani, M. (2016). Scots: A tool for the synthesis of symbolic controllers, *19th International Conference on Hybrid Systems: Computation and Control, HSCC'16, Vienna, Austria*, pp. 99–104.

Saoud, A. (2019). *Compositional and Efficient Controller Synthesis for Cyber-Physical Systems*, PhD thesis, Université Paris-Saclay, Gif sur Yvette.

Saoud, A., Girard, A. and Fribourg, L. (2019). Assume-guarantee contracts for discrete and continuous-time systems, *Preprint*, `https://hal.archives-ouvertes.fr/hal-02196511`.

Saoud, A., Girard, A. and Fribourg, L. (2020). Contract-based design of symbolic controllers for safety in distributed multiperiodic sampled-data systems, *IEEE Transactions on Automatic Control*, DOI:10.1109/TAC.2020.2992446.

Saoud, A., Jagtap, P., Zamani, M. and Girard, A. (2018). Compositional abstraction-based synthesis for cascade discrete-time control systems, *6th IFAC Conference on Analysis and Design of Hybrid System, Oxford, UK*.

Tabuada, P. (2009). *Verification and Control of Hybrid Systems*, Springer, New York, NY.

Tazaki, Y. and Imura, J. (2012). Discrete abstractions of nonlinear systems based on error propagation analysis, *IEEE Transactions on Automatic Control* **57**(3): 550–564.

Weber, A., Rungger, M. and Reissig, G. (2017). Optimized state space grids for abstractions, *IEEE Transactions on Automatic Control* **62**(11): 5816–5821.

Wongpiromsarn, T., Topcu, U., Ozay, N., Xu, H. and Murray, R. (2011). Tulip: A software toolbox for receding horizon temporal logic planning, *14th International Conference on Hybrid Systems: Computation and Control, HSCC'11, Chicago, IL, USA*, pp. 313–314.

Zamani, M., Esfahani, P.M., Majumdar, R., Abate, A. and Lygeros, J. (2014). Symbolic control of stochastic systems via approximately bisimilar finite abstractions, *IEEE Transactions on Automatic Control* **59**(12): 3135–3150.

Zamani, M., Pola, G., Mazo, M. and Tabuada, P. (2012). Symbolic models for nonlinear control systems without stability assumptions, *IEEE Transactions on Automatic Control* **57**(7): 1804–1809.

Zhai, G., Chen, N. and Gui, W. (2013). Decentralized design of interconnected H_∞ feedback control systems with quantized signals, *International Journal of Applied Mathematics and Computer Science* **23**(2): 317–325, DOI:10.2478/amcs-2013-0024.

Zolghadri, A., Henry, D. and Monsion, M. (1996). Design of nonlinear observers for fault diagnosis: A case study, *Control Engineering Practice* **4**(11): 1535–1544.

Zonetti, D., Saoud, A., Girard, A. and Fribourg, L. (2019). A symbolic approach to voltage stability and power sharing in time-varying DC microgrids, *2019 18th European Control Conference (ECC), Naples, Italy*, pp. 903–909.

W. Alejandro Apaza-Perez received his MSc degree in mathematics (2014) and his PhD (with honors) in electrical engineering and automatic control (2018), both from the National Autonomous University of Mexico (UNAM) in Mexico City. He also holds a Bachelor's degree in mathematics (2011). Between 2018 and 2020, he was a member of the ARIA team in the Control System Group of the IMS Lab (Integration: from Material to Systems), Bordeaux, France. Currently, he is a postdoctoral researcher at the Laboratory of Signals and Systems (L2S), Gif-sur-Yvette, France. His present research covers computational approaches, formal methods and applications to cyber-physical systems. He is also interested in robust and nonlinear control/observers, dissipative systems, high order sliding mode control, and their applications.

Christophe Combastel received his MSc degree in electrical engineering (1997) and his PhD in control systems (2000), both from the Grenoble Institute of Technology (Grenoble-INP), France. From 2001 to 2015, he was an associate professor at ENSEA. Since 2015, he has been with the University of Bordeaux and the IMS Lab (CNRS UMR5218), France. His main focus is on dynamic model-based decision-making in uncertain contexts. As a member of the ARIA team in the Control System Group of IMS, his research interests include interval, set-membership and stochastic algorithms for integrity control applications ranging from on-line fault diagnosis to verification and synthesis of cyber-physical systems, with special emphasis on uncertainty propagation and model-based data fusion.

Ali Zolghadri is currently an exceptional class university professor in control and system engineering at the University of Bordeaux, France. His research deals with model-based fault management issues. More recently, his research interests have included interconnected, hybrid and distributed engineered systems by combining aspects of symbolic methods/models and robust techniques from control theory, and new methods for autonomous navigation and safety-related issues in future civil aviation operations. He has authored and co-authored about 80 papers in leading international journals, about 140 communications in international conferences, one Springer book and 12 book chapters. He is a co-holder of 15 patents (French and US) in the aerospace field. He has been the coordinator of a number of collaborative French, European and international research projects and actions in control and aeronautics. He has coordinated (or participated in) a number of collaborative French, European (FP7, H2020) and international (Mexico, the USA, China and Russia) research projects. He has developed with his team the first model-based fault management system for new generation A350 aircraft, which entered commercial service in 2015 and is now flying worldwide. He is the recipient of the 2016 CNRS Innovation Medal for outstanding scientific research with innovative applications in the technological and societal fields.

Received: 3 February 2020
Revised: 19 July 2020
Re-revised: 18 August 2020
Accepted: 26 August 2020