ON DISTRIBUTED SYMBOLIC CONTROL OF INTERCONNECTED SYSTEMS UNDER PERSISTENCY SPECIFICATIONS

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

aIMS 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., 2020; 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 ♦□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
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 $R(A_0) = \{ b \in B \mid \exists a \in A_0 \text{ such that } (a, b) \in R \}$. $f : A \to B$ denotes an ordinary map and $f^{-1}(b) := \{ a \in A : f(a) = b \}$ for $b \in B$. The symbols $\mathbb{R}$, $\mathbb{Z}_{>0}$, $\mathbb{N}_0$ denote the set of real numbers, positive real numbers, integers, natural numbers including the zero, respectively. $[a; b] \subseteq \mathbb{Z}$ 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: $J = [1; n] \subseteq \mathbb{N}$, $\hat{J} = J \setminus \{1\}$. Given two sets $A$ and $B$, the product $A \times B$ denotes the Cartesian product, and for a collection of sets $\{ A_j \}_{j \in \hat{J}}$, indexed by the set $\hat{J}$, the product $\prod_{j \in \hat{J}} A_j$ denotes the Cartesian product keeping the order in $\hat{J}$, i.e., $\prod_{j \in \hat{J}} A_j := A_1 \times \cdots \times A_{i-1} \times A_{i+1} \times \cdots \times A_n$. Given a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, $x_i$ denotes the $i$-th component of $x$, and $\bar{x}_i$ is defined as $\bar{x}_i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

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 input $u$ if and only if $(x, u, x') \in F_s$. A transition $(x, u, x') \in F_s$ is also denoted by $x \trans{u} x'$. $\text{Post}_a(x) = \{ x' \in X_s : (x, u, x') \in F_s \}$ 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:

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

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: $\text{Post}_{u_b}(x_b)$ 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:

\[ \dot{\xi}(t) = g(\xi(t), v(t)) + w(t), \]

with state variable \( \xi \in X \subset \mathbb{R}^n \), control input \( v \in U \subset \mathbb{R}^m \), disturbance \( w \in [W_{\min}, W_{\max}] \) which denotes a hyper-interval \([W_{\min,1}, W_{\max,1}] \times \cdots \times [W_{\min,n}, W_{\max,n}]\)

The above structure results from the interconnection of the 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 \( \forall i \in \mathcal{I}, x_i(k, x_{0i}, u_i) \) with initial condition \( x_i(0, x_{0i}, u_i) = x_{0i} \), discrete time \( k \in \mathbb{N}_0 \), and control \( c_i : X_i \times \prod_{j \in \mathcal{J}_i} X_j \to U_i \) where the domain of the controller is defined as \( \text{dom}(c) = \{x \in X | c(x) \neq \emptyset\} \).

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

\[ \diamond P : \forall x_0 \in \text{dom}(c), \exists k_i \in \mathbb{N}_0, x_i(k_i, x_{0i}, u_i) \in P_i, \]

\[ \Box P : \forall x_0 \in \text{dom}(c), \exists k_i \in \mathbb{N}_0, \forall k \geq k_i, \]

\[ x_i(k, x_{0i}, u_i) \in P_i. \]

The objective is to design a controller \( c_i \) for the subsystem \( i \) in a domain \( X_i \times \prod_{j \in \mathcal{J}_i} D_j \) of smaller cardinality than \( X_i \times \prod_{j \in \mathcal{J}_i} X_j \) 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

\[ F_S = \{ (x, u, x') \in X_S \times U_S \times X_S : \exists i \in \mathcal{I}, x_i' \in f_i(x_i, \tilde{x}_i, u_i) \}. \]

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

\[ V_i : X_i \to D_i \]

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

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

\[ F_i(x_i, \tilde{x}_i, u_i) = \bigcup_{\tilde{x}_i \in \prod_{j \in \mathcal{J}_i} V_i^{-1}(u_i)} f_i(x_i, \tilde{x}_i, u_i). \]

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

\[ CPre^{\tilde{v}_i} (S_i|U_i) = \{ x_i \in X_i : \exists u_i \in U_i, F_i(x_i, \tilde{v}_i, u_i) \subseteq S_i \}. \]

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 $V^+_i(v_i, \tilde{v}_i)$ for all $i \in J$ defined with the
controllable predecessor:

$$V^+_i(v_i, \tilde{v}_i) = \min \left\{ k \in \mathbb{N}_0 : V^{-1}_i(v_i) \subseteq CPr_{\tilde{e}_i} V^+_i (V^+_i, \leq k) \mid U_i \right\}. \quad (7)$$

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

$$F_T = \left\{ (v, u_T, v') \in X_T \times \{ u_T \} \times X_T : \forall i \in J, v'_i \leq V^+_i (v_i, \tilde{v}_i) \right\}. \quad (8)$$

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

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

is an alternating simulation relation from $T$ to $S$.

The next result gives the domain of admissible controllers for $T$ satisfying the persistance 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 $c = (c_1, \ldots, c_n)$, where $c_i$ has domain $X_i \times \prod_{j \in J} D_j$ enforcing the specification $\Diamond \Box P_S$ given by

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

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

$$c_i (x_i, \tilde{v}_i) \in \left\{ u_i \in U_i : \max_{\tilde{x}_i \in \prod_{j \in J} D_j} V_i (f_i (x_i, \tilde{x}_i, u_i)) \leq V^+_i (V_i (x_i), \tilde{v}_i) \right\}. \quad (11)$$

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

The system $T$ defined by (7) and (8) can be considered as a directed graph $G = (X_T, E_T)$, where $(v, v') \in E_T$ 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 $G = (X_T, E_T)$ are included in $P_T$.

**Proof.** (Necessity) Assume that $T$ satisfies $\Diamond \Box P_T$ and consider a cycle $v(1), v(2), \ldots, v(m) = v(1)$ that is reachable from some initial state $v(1)$, through a path $v(1) \rightarrow v(2) \rightarrow \cdots \rightarrow v(k) = v(1)$. Since the infinite path $v(1) \rightarrow v(2) \rightarrow \cdots \rightarrow v(m - 1) \rightarrow \cdots$ 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 $P_T = P_T \cap \cdots \cap P_n T \subseteq X_T$. 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 $(v(1), \ldots, v(n)) \in P_T$, if $w(v_i) \leq v(i)$ for all $i \in J$, then $(w(1), \ldots, w(n)) \in P_T$. $T$ satisfies the specification $\Diamond \Box P_T$ if the following conditions are satisfied:

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

(ii) $\forall v \in X_T \setminus P_T, \forall i \in J, V^+_i (v_i, \tilde{v}_i) \leq v_i$,

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

**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 $P_T \setminus X_T \setminus P_T$ 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 $v(2) \leq V^+_1 (v_1 (1), \tilde{v}_1 (1))$ and $v_1 (3) \leq V^+_1 (v_1 (2), \tilde{v}_1 (2))$. 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 \( v(1) \xrightarrow{u_T} v(2) \xrightarrow{u_T} \ldots \xrightarrow{u_T} v(m) \) for arbitrary values of \( m \), proving \( v(1) = v(2) = \ldots = v(m) \).

**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 \( (v, \{ u_T \}, v) \in E_T \). Using the relation (8), it is equivalent to \( v_i \leq V(v_i, u_i) \) for all \( i \in I \), this is a contradiction with the condition (iii).

These claims guarantee that all cycles are included in \( P_T \). Thus, \( T \) satisfies the specification \( \diamond \square P_T \) by Proposition I.

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 \( \{V_i\}_{i \in I} \) 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, \( E \subseteq \prod_{j \in J} X_j \) and \( S_i \subseteq X_i \), is defined by

\[
CP_i(U_i, E, S_i) = \left\{ x_i \in X_i : \exists u_i \in U_i, \forall \bar{x}_i \in E, f_i(x_i, \bar{x}_i, u_i) \subseteq S_i \right\}.
\]

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 \( \{V_i\}_{i \in I} \) are given, and \( v \in \prod_{i \in I} D_i \) (the domain of ranking functions), then taking \( E = \prod_{j \in J} V_j^{-1}(v_j) \) allows us to recover the predecessor defined in (6), through

\[
CP_i \left( U_i, \prod_{j \in J} V_j^{-1}(v_j), S_i \right) = CPre^{\hat{v}_i}_i \left( S_i | U_i \right). \quad (14)
\]

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) \( E_{i_1} \subseteq E_{i_2} \Rightarrow CP_i \left( U_i, E_{i_2}, S_i \right) \subseteq CP_i \left( U_i, E_{i_1}, S_i \right) \)

(ii) \( S_{i_1} \subseteq S_{i_2} \Rightarrow CP_i \left( U_i, E_{i_1}, S_{i_1} \right) \subseteq CP_i \left( U_i, E_{i_1}, S_{i_2} \right) \).

**Proof.**

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

\[ \exists u_i \in U_i, \forall \bar{x}_i \in E_{i_1}, f_i(x_i, \bar{x}_i, u_i) \subseteq S_i, \]

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

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

\[ \exists u_i \in U_i, \forall \bar{x}_i \in E_{i_1}, f_i(x_i, \bar{x}_i, u_i) \subseteq S_{i_2}, \]

which implies \( x_i \in CP_i \left( U_i, E_{i_1}, S_{i_2} \right) \) by (13).

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

The idea of Algorithm I 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 (14) reaches lower levels, which is denoted \( Z_i( \leq k - 1 ) \), despite the influence of the sets \( Z_j(\ell_s) \) 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. I 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 I, 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 $Z_i(k) \subseteq X_i$, $i \in I$, $k \in \mathbb{N}_0$ and controller domain.

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

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

Step 2. Find sets $B_i \subseteq P_i, \forall \, i \in I$ such that

$$B_i \subseteq CP_i \left( U_i, \prod_{s \in J_i} Z_s(0), B_i \right). \quad (16)$$

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

$$k \leftarrow 0; \quad \text{for } i \leftarrow 1, n \quad \text{do } Z_i(0) \leftarrow B_i \quad \text{end for;}
\quad \text{while } \exists i \in I, \forall \, i \in I
\quad \text{do } k \leftarrow k + 1;
\quad \text{for } i \leftarrow 1, n \quad \text{do}
\quad \text{for } L(i) \leftarrow \{0, 1, \ldots, k\}^{(i-1) \times \{k\} \times \{0, 1, \ldots, k-1\}^{(n-i)}; \quad (17)
\quad \text{cond} = \forall \ell \in L(i), \forall \, j \in I, \forall q \in \{k - b \mid j > i\}, Z_j(q) \subseteq CP_j \left( U_j, \prod_{s \in J_i} Z_s(\ell_s), Z_j(\leq q) \right); \quad (18)
\quad \text{if cond } \wedge \text{not } (Z_i(\leq k - 1), \text{then})
\quad Z_i(k) \leftarrow Z_i(k) \setminus Z_i(\leq k - 1); \quad (19)
\quad \text{else}
\quad Z_i(k) \leftarrow \emptyset;
\quad \text{end if}
\quad \text{end for}
\quad \text{end for}
\quad \text{end while}$$

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 $P_i$” is fulfilled (see also Theorem 2).

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

$$\text{dom}(c) = \prod_{i \in I} Z_i(\leq k_{\text{max}}(i)), \quad (20)$$

where $\forall \, i \in I, k_{\text{max}}(i) := \max\{k \in \mathbb{N}_0 \mid Z_i(k) \neq \emptyset\}$. Note that, $\emptyset$ denoting the empty set (to be distinguished from a non-empty “zero” value) and provided $f_i(x_i, \bar{x}_i, u_i)$ 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).

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

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

Let $X_i^Z := \bigcup_{k \in [0,d_i]} Z_i(k)$ be the domain where the ranking functions are defined, and $X_i^Z = \prod_{i \in I} X_i^Z$. $X_i^Z = \prod_{i \in I} \{0, 1, \ldots, d_i\}$ denotes the state space used to build the reduced abstraction according to Section 3.

Lemma 3. Consider the sets $Z_i$ obtained from Algorithm 7 The ranking functions $V_i$ defined in $X_i^Z$ as

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

satisfy

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

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 7 Define the ranking functions $V_i : X_i^Z \subseteq X_i \rightarrow \mathbb{N}_0, \forall \, i \in I$ based on the sets $Z_i$ as in (22) and define a system $\bar{T}$ as in (7) and (8). Let

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 $Z_i$’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.
Claim 2. The inclusion (26) ensures (27) according to (7):

In terms of ranking functions, Step 2 in Algorithm 1 implies

\[ \forall i \in J, \quad V_i^{-1}(0) = P_i, \quad (24) \]

which implies \( P_T = \{0\} \) where \( 0 \in X_T \) is the zero vector.

In terms of ranking functions, Step 2 in Algorithm 1 implies

\[ \forall i \in J, \quad V_i^{-1}(0) \subseteq CP_i(U_i, \prod_{s \in J} V_s^{-1}(0), V_i^{-1}(0)). \quad (25) \]

Note that \( CP_i(U_i, \prod_{s \in J} V_s^{-1}(0), V_i^{-1}(0)) = CP_{\tilde{J}}(V_i^{-1}(0)|U_i) \) is satisfied by (14), from which the inclusion (25) can be expressed as

\[ \forall i \in J, \quad V_i^{-1}(0) \subseteq CP_{\tilde{J}}(V_i^{-1}(0)|U_i). \quad (26) \]

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

\[ \forall i \in J, \quad V_i^{+}(0) = 0. \quad (27) \]

Claim 2. \( \forall v \in X_T^Z \setminus P_T, \forall i \in J, \quad V_i^{+}(v_i, \bar{v}_i) \leq v_i. \)

Let \( v \in X_T^Z \setminus P_T \), then \( \forall i \in J, \quad \forall i \in [0; d] \).

\[ \bar{v}_m = \max \{ v_i : i \in J \}, \]

\[ r = \max \{ i \in J : v_i = \bar{v}_m \}, \quad (28) \]

which implies that \( v_i \leq \bar{v}_m \) for \( i \leq r \), and \( v_i < \bar{v}_m \) for \( i > r \).

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

\[ T_r = Z_r(v_r), \]

\[ C \cap \bigcup_{\ell \in L(r)} CP_r(U_r, \prod_{s \in \tilde{J}} Z_s(\ell_s), Z_r(\leq v_r - 1)), \quad (29) \]

\[ L(r) = \{0, 1, \ldots, \bar{v}_m\}^{(r-1)} \times \{\bar{v}_m\} \]

\[ \times \{0, 1, \ldots, \bar{v}_m - 1\}^{(n-r)}, \]

\[ Z_j(v_j) \subseteq CP_j \left( U_j, \prod_{s \in \tilde{J}} Z_s(v_s), Z_j(\leq v_j) \right). \quad (30) \]

\[ \forall j \in \tilde{J}, \text{by (17) and (18).} \]

Lemma 3 implies \( V^{-1}(\leq v_i) = Z_i(\leq v_i) \) and \( \prod_{s \in \tilde{J}} V_s^{-1}(v_s) = \prod_{s \in \tilde{J}} Z(v_s), \) ensuring

\[ \forall j \in \tilde{J}, \quad CP_j \left( U_j, \prod_{s \in \tilde{J}} Z_s(v_s), Z_j(\leq v_j) \right) = CP_j \left( U_j, \prod_{s \in \tilde{J}} V_s^{-1}(v_s), V_j^{-1}(\leq v_j) \right). \quad (31) \]

For each \( j \in \tilde{J}, \) we have that \( V_j^{-1}(v_j) = Z_j(v_j) \) and \( CP_j(U_j, \prod_{s \in \tilde{J}} V_s^{-1}(v_s), V_j^{-1}(\leq v_j)) = CP_{\tilde{J}}(Z_j^{-1}(\leq v_j)|U_j) \) are obtained by Lemma 3 and (14) respectively, which imply

\[ \forall j \in \tilde{J}, \quad V_j^{-1}(v_j) \subseteq \prod_{s \in \tilde{J}} CP_{\tilde{J}}(Z_j^{-1}(\leq v_j)|U_j). \quad (32) \]

from (30) and (31). Consequently, \( V_j^{+}(v_j, \bar{v}_j) \leq v_j, \forall j \in \tilde{J} \) is satisfied from (7).

Moreover, the inclusion

\[ \bigcap_{\ell \in L(r)} CP_r(U_r, \prod_{s \in \tilde{J}} Z_s(\ell), Z_r(\leq v_r - 1)) \]

\[ \subseteq CP_r \left( U_r, \prod_{s \in \tilde{J}} Z_s(v_s), Z_r(\leq v_r - 1) \right), \quad (33) \]

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

\[ CP_r \left( U_r, \prod_{s \in \tilde{J}} Z_s(v_s), Z_r(\leq v_r - 1) \right) \]

\[ \subseteq CP_r \left( U_r, \prod_{s \in \tilde{J}} V_s^{-1}(v_s), V_r^{-1}(\leq v_r) \right). \quad (34) \]

The following relation is obtained from (14):

\[ CP_r \left( U_r, \prod_{s \in \tilde{J}} V_s^{-1}(v_s), V_r^{-1}(\leq v_r) \right) = \]

\[ CP_{\tilde{J}}(V_j^{-1}(\leq v_j)|U_j). \quad (35) \]

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

\[ V_j^{+}(v_j, \bar{v}_j) < v_r. \quad (36) \]

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

\[ \forall v \in X_T^Z \setminus P_T, \forall i \in J, \quad V_i^{+}(v_i, \bar{v}_i) \leq v_i. \quad (37) \]

Claim 3. \( \forall v \in X_T^Z \setminus P_T, \exists i \in J, \quad V_i^{+}(v_i, \bar{v}_i) < v_i. \)
Claim 3 is guaranteed from a process analogous to that of case \( i = r \) in Claim 2, where (33), (29), (34) and (35) imply 
\[
V_r^w(\bar{v}_r, \bar{u}_r) < v_r.
\]

Claims 1–3 satisfy the conditions of Proposition 2, which imply that the system \( T \), according to (7) and (8), enforces the specification \( \Box \Box \mathcal{R} \) in \( X \mathcal{Y} \).

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 \( q_{\text{max}} \) 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
\[
\begin{align*}
\dot{h}_1(t) &= \frac{1}{S}(q_1 - a_{z1}\text{sign}(h_1 - h_2)\sqrt{2g(h_1 - h_2)}), \\
\dot{h}_2(t) &= \frac{1}{S} \left( S_{a_{z2}}\text{sign}(h_2 - h_3)\sqrt{2g(h_2 - h_3)} - S_{a_{z2}}\text{sign}(h_2 - h_3)\sqrt{2g(h_2 - h_3)} \right), \\
\dot{h}_3(t) &= \frac{1}{S} \left( q_2 - a_{z3}\text{sign}(h_2 - h_3)\sqrt{2g(h_2 - h_3)} \right).
\end{align*}
\tag{38}
\]

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

The linearized state-space model in continuous form is
\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t). 
\end{align*}
\tag{39}
\]

The linearization technique is valid in the vicinity of an operating point. The nonlinear system \( \ref{eq:38} \) is linearized around the following steady state operating point: \( [h_{1o} \ h_{2o} \ h_{3o}]^T = [0.6 \ 0.5 \ 0.4]^T \) m and \( [q_{1o} \ q_{2o}]^T = [0.35787 \ 0.6563]^T \times 10^{-4} \text{ m}^3/\text{s}. \)

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tank cross-section area</td>
<td>( S = 0.0171 ) m(^2)</td>
</tr>
<tr>
<td>Pipe cross-section area</td>
<td>( S_c = S_1 = S_2 = S_3 ) = 0.00005 m(^2)</td>
</tr>
<tr>
<td>Pipe outflow coefficients</td>
<td>( a_{z1} = 0.511 ), ( a_{z2} = 0.5279 ), ( a_{z3} = 0.7313 )</td>
</tr>
<tr>
<td>Maximum level</td>
<td>( h_{\text{max}} = 0.68 ) m</td>
</tr>
<tr>
<td>Maximum inflow rate</td>
<td>( q_{\text{max}} = 1.2 \times 10^{-4} ) m(^3)/s</td>
</tr>
</tbody>
</table>

In the first step, the tool SCOTS (Rungger and Zamani, 2016) is used to abstract the infinite state system from \( \ref{eq:38} \) as a finite transition system. SCOTS constructs non-deterministic transition systems, which are obtained using a state-quantization parameter \( \eta \), an input-quantization parameter \( \mu \), and a time-quantization parameter \( \tau \). Three admissible values for the control signals \( q_1 \)'s have been selected around the operating point: \( \forall i \in \{1, 2\}, U_i = \left\{ q_{i\ell o}, \frac{1}{2}(q_{i\ell o} + \eta), q_{i\ell o}, \frac{1}{2}(q_{i\ell o} - \eta) \right\} \) with \( q_{i\ell o} = 0.1 \times 10^{-4} \), \( q_{i\ell o} = 0.7 \times 10^{-4} \), \( q_{i\ell o} = 0.3 \times 10^{-4} \), \( q_{i\ell o} = 0.9 \times 10^{-4} \). The quantization parameters are considered as follows: \( \forall i \in \{1, 2\}, j \in \{1, 2, 3\}, \eta_j = 0.005, \mu = \frac{\eta}{2}(q_{i\ell o} - q_{i\ell o}) \) and \( \tau = 0.1 \) s. The transition relation \( \ref{eq:38} \) 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: \( 0.59 \leq h_1 \leq 0.61, 0.48 \leq h_2 \leq 0.52, 0.39 \leq h_3 \leq 0.41. \)
Thus, Algorithm 1 is initialized at Step 1 (see (15)) with the discrete target sets given by
\[ P_1 = \eta_1 Z \cap [0.59, 0.61], \]
\[ P_2 = \eta_2 Z \cap [0.48, 0.52], \]
\[ P_3 = \eta_3 Z \cap [0.39, 0.41]. \]
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).

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


switching control, in K.G. Larsen et al. (Eds), Reachability Problems, Springer, Cham, pp. 119–133.


On distributed symbolic control of interconnected systems under persistency specifications


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.

Accepted: 26 August 2020