Automatica 95 (2018) 368–377

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

What information really matters in supervisor reduction?☆

Rong Su [a,*], W. Murray Wonham [b]

[a] School of Electrical & Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
[b] The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at University of Toronto, 10 King's College Road, Toronto, Ontario, M5S 3G4, Canada

ARTICLE INFO

Article history:
Received 2 October 2016
Received in revised form 26 December 2017
Accepted 6 May 2018
Available online 18 June 2018

Keywords:
Discrete-event systems
Supervisory control
Supervisor reduction
Control equivalence
Partial observation

ABSTRACT

To make a supervisor comprehensible to a designer has been a long-standing goal in the supervisory control community. One strategy is to reduce the size of a supervisor to generate a control equivalent version, whose size is optimistically much smaller than the original one so that a user or control designer can easily check whether a designed controller fulfils its objectives and requirements. After the first journal paper on this topic appeared in 1986 by Vaz and Wonham, which relied on the concept of control covers, Su and Wonham proposed in 2004 to use control congruences to ensure computational viability. This work was later adopted in supervisor localization theory, which aims for a control equivalent distributed implementation of a given centralized supervisor. Despite these publications some fundamental questions, which might have been addressed in the first place, have not yet been answered, namely what information is critical to ensure control equivalence, what information is responsible for size reduction, and whether partial observation makes the problem essentially different. In this paper we address these questions by showing that there exists a unified supervisor reduction theory, which is applicable to all feasible supervisors regardless of whether they are under full observation or partial observation. Our theory proposes a preorder (called *leanness*) over all control equivalent feasible supervisors based on their enabling, disabling and marking information such that, if a supervisor S_1 is leaner than another supervisor S_2, then the size of the minimal control cover defined over the state set of S_1 is no bigger than that of S_2.

1. Introduction

In supervisory control theory (SCT) (Ramadge & Wonham, 1987; Wonham & Ramadge, 1987), the control problem associated with a discrete-event system (DES) is to enforce controllable and nonblocking behavior of the plant that is admissible by the specification. When applying SCT to a real application, there are two basic questions that require a user to answer, that is, are we doing the right thing, and are we doing things in the right way. The first question is about the correctness of the plant and requirement models. The second is about correctness of supervisor synthesis, which, if computational complexity is not a concern, has been adequately answered by SCT researchers. When computational complexity is indeed a concern, several efficient synthesis approaches have been proposed in the literature, e.g., Feng and Wonham (2008), Mohajerani, Malik, and Fabian (2014), Su, Van Schuppen, and Rooda (2010, 2012), which can ensure correct behaviors of the closed-loop system with low computational complexity. The first question, on the other hand, has been a long-standing hurdle to SCT being adopted by industry because so far there is no efficient way to identify potential errors in plant models or requirement models. The current practice is to synthesize a supervisor based on a given plant model and requirements. An empty supervisor usually indicates a fault either in the model or in the requirements; this should prompt the system designer to undertake model or requirement updates. The current SCT and its relevant tools can assist the designer to quickly locate the problems in the model that lead to emptiness of the supervisor. The real challenge is how to determine whether the plant model and the requirements are correct, when the supervisor synthesis returns a non-empty supervisor. In this case it usually requires not only syntactic correctness but also semantic correctness, i.e., the designer has to understand the true meaning and impact of every transition in the synthesized supervisor. Thus, to make a supervisor small enough for a designer to understand its function becomes important.

A supervisor carries two types of information: the key information at each state for event enabling/disabling and marking,

☆ The funding supports from Singapore Ministry of Education Tier 1 Academic Research Grant M4011221.040 RG84/13 (2013-T1-002-177), Singapore National Research Foundation Delta-NTU Corporate Lab Program (SMA-RP2), and from the Natural Sciences and Engineering Research Council (NSERC) of Canada, Grant DG7399, are gratefully acknowledged. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: rsu@ntu.edu.sg (R. Su), wonham@ece.utoronto.ca (W.M. Wonham).

https://doi.org/10.1016/j.automatica.2018.06.004
0005-1098/

and the information that tracks the evolution of the plant. The latter may contain some redundancy because the plant itself also carries such evolution information. In principle, it is possible to remove redundant transitional information from the supervisor, which will not interfere with the first kind of information, i.e., a reduced supervisor can still ensure the same control capability as that of the original supervisor. This is the key idea used in Vaz and Wonham's paper on supervisor reduction (Vaz & Wonham, 1986), which relies on the concept of *control cover*. A control cover is a collection of subsets of states in a supervisor, in which the states of each subset are "control consistent" with respect to event enabling/disabling and state marking; the exact meaning will be explained later. The authors proved two reduction theorems, and proposed a corresponding (exponential time) reduction algorithm. To overcome the computational complexity involved in supervisor reduction, Su and Wonham made a significant extension in Su and Wonham (2004) by first relaxing the concept of control cover, then providing a polynomial-time reduction algorithm based on a special type of cover called *control congruence*, and finally showing that the minimal supervisor problem (MSP) of computing a supervisor with minimal state size is NP-hard. A polynomial-time lower bound estimation algorithm provided in Su and Wonham (2004) has indicated that in many instances minimal supervisors can be achieved in polynomial time by using control congruence. Since then, this reduction algorithm has been used with gratifying results. One major application of supervisor reduction is in supervisor localization (Cai & Wonham, 2010), which aims to create a control-equivalent distributed implementation of a given centralized supervisor.

The supervisor reduction theory proposed in Su and Wonham (2004) rests on two basic assumptions: (1) only full observation is considered; (2) the supervisor under consideration represents a sublanguage of the plant, which can be easily satisfied by applying supremal synthesis. Since then, many questions have been raised by users. For example, can we apply supervisor reduction to partially reduced supervisors (which may not necessarily represent sublanguages of a given plant) and can we apply supervisor reduction in cases with partial observation? Some results have been reported in the literature about the second question, see e.g., Zhang and Cai (2016). The main objective of supervisor reduction is to ensure control equivalence between the original supervisor and a reduced supervisor. The fundamental questions are (1) **Q1:** what information ensures control equivalence, even under partial observation, and (2) **Q2:** what information determines the state size of a reduced supervisor, which is the main performance index of supervisor reduction. Ever since (Su & Wonham, 2004; Vaz & Wonham, 1986), these questions are still open. In this paper we aim to provide an answer. We first propose a generalized supervisor reduction theory which is applicable to all *feasible* supervisors, regardless of whether they are subject to full observation or partial observation; here a feasible supervisor does not disable uncontrollable events and always issues the same control command after strings that are not distinguishable based on observations. In the case of partial observation, a supervisor does not in general represent a sublanguage of the plant. We show that for each feasible supervisor S of a plant G, there exists a feasible supervisor **SUPER** derivable from the linguistic definition of *uncertainty subset construction* (Wonham, 2016). **SUPER** has the "universal" property that any feasible supervisor that is control equivalent to S with respect to G, and non-redundant with respect to S (i.e. without superfluous transitions), can be projected from **SUPER** via a suitable control cover on its state space, namely is a "quotient" of **SUPER** with respect to this cover. This result will answer our first question **Q1**. After that, we define a preorder $\preceq$ (referred to as "leanness") on feasible supervisors by using key information about event enabling/disabling and state marking such that for any two control equivalent supervisors S_1 and S_2 with respect to G, if S_1 is leaner than S_2, i.e., $S_1 \preceq S_2$, then the minimal reduced supervisor induced by a minimal control cover on S_1 is no bigger than the one induced by a minimal control cover on S_2. This result provides an answer to the second question **Q2**. As a direct consequence of this result, as long as control equivalence holds, a feasible supervisor under full observation always results in a reduced supervisor no bigger than the one induced from a supervisor under partial observation. Our theory is independent of the specific way of achieving observability, for instance via the property of normality (Lin & Wonham, 1988) or of relative observability (Cai, Zhang, & Wonham, 2013), or by direct search (Lin & Wonham, 1988) — the effect of such a choice is lumped into the property of control feasibility, which states that a feasible supervisor must apply the same control law to all transitional sequences which cannot be distinguished based on observations.

The remainder of the paper is organized as follows. In Section 2, we provide preliminaries on supervisor reduction. In Section 3 we discuss critical information for ensuring control equivalence. Then in Section 4 we address information that determines reduction efficiency. We draw conclusions in Section 5.

2. Preliminaries on supervisor reduction

Given an arbitrary finite alphabet Σ, let Σ^* be the free monoid on Σ whose elements are all the finite strings of zero or more elements from Σ, with the empty string ϵ being the identity element and string concatenation being the monoid operation. Given two strings $s, t \in \Sigma^*$, s is called a *prefix substring* of t, written as $s \leq t$, if there exists $s' \in \Sigma^*$ such that $ss' = t$, where ss' denotes the concatenation of s and s'. Any subset $L \subseteq \Sigma^*$ is called a *language*. The *prefix closure* of L is defined as $\overline{L} = \{s \in \Sigma^* | (\exists t \in L) s \leq t\} \subseteq \Sigma^*$. Given two languages $L, L' \subseteq \Sigma^*$, let $LL' := \{ss' \in \Sigma^* | s \in L \wedge s' \in L'\}$ denote their concatenation. Let $\Sigma' \subseteq \Sigma$. A mapping $P : \Sigma^* \to \Sigma'^*$ is called the *natural projection* with respect to (Σ, Σ'), if

(1) $P(\epsilon) = \epsilon$,
(2) $(\forall \sigma \in \Sigma) P(\sigma) := \begin{cases} \sigma & \text{if } \sigma \in \Sigma', \\ \epsilon & \text{otherwise}, \end{cases}$
(3) $(\forall s\sigma \in \Sigma^*) P(s\sigma) = P(s)P(\sigma)$.

Given a language $L \subseteq \Sigma^*$, $P(L) := \{P(s) \in \Sigma'^* | s \in L\}$. The inverse image mapping of P is

$$P^{-1} : 2^{\Sigma'^*} \to 2^{\Sigma^*} : L \mapsto P^{-1}(L) := \{s \in \Sigma^* | P(s) \in L\},$$

where $2^{\Sigma'^*}$ and 2^{Σ^*} denote the power sets of Σ'^* and Σ^*, respectively. When L is a singleton, say $L = \{s\}$, we will using $P^{-1}(s)$ to denote $P^{-1}(\{s\})$ for simplicity throughout the paper.

A plant is modeled as a *deterministic finite-state automaton*, $G = (X, \Sigma, \xi, x_0, X_m)$, where X stands for the state set, Σ for the alphabet, $\xi : X \times \Sigma \to X$ for the (partial) transition function, x_0 for the initial state and $X_m \subseteq X$ for the marker state set. Here we adopt the notation in Wonham (2016) and write $\xi(x, \sigma)!$ to denote that the transition $\xi(x, \sigma)$ is defined. The domain of ξ can be extended to $X \times \Sigma^*$, where $\xi(x, \epsilon) = x$ for all $x \in X$, and $\xi(x, s\sigma) := \xi(\xi(x, s), \sigma)$. The *closed* behavior of G is defined as $L(G) := \{s \in \Sigma^* | \xi(x_0, s)!\}$, and the *marked* behavior of G is $L_m(G) := \{s \in L(G) | \xi(x_0, s) \in X_m\}$. G is *nonblocking* if $\overline{L_m(G)} = L(G)$. We say G is *reachable* if for each $x \in X$ there exists $s \in L(G)$ such that $\xi(x_0, s) = x$. From now on we consider only reachable automata. We denote by $|X|$ the size of the state set X. In some circumstances, when the state set is not explicitly mentioned, we also write $|G|$ for the size of an automaton, namely the size of its state set. Given two finite-state automata $G_i = (X_i, \Sigma, \xi_i, x_{i,0}, X_{i,m})$ $(i = 1, 2)$, the *meet* of G_1 and G_2, denoted as $G_1 \wedge G_2$, is a (reachable) finite-state automaton such that $L(G_1 \wedge G_2) = L(G_1) \cap L(G_2)$ and $L_m(G_1 \wedge G_2) = L_m(G_1) \cap L_m(G_2)$.