Information Sciences 465 (2018) 523–544

Contents lists available at ScienceDirect

Information Sciences

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

Supervisory control of state-tree structures with partial observation[☆]

Chan Gu[a], Xi Wang[a], Zhiwu Li[b,c,*], Naiqi Wu[b]

[a] School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[b] Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau
[c] Key Laboratory of Electronic Equipment Structure Design, Ministry of Eduction, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 20 May 2017
Revised 13 February 2018
Accepted 8 July 2018

Keywords:
Discrete-event system
Supervisory control
State-tree structure
Observability
Predicate

ABSTRACT

Supervisory control of discrete-event systems (SCDES) is well developed to find a maximally permissive supervisor. As an extension to supervisory control theory, a new framework, state-tree structures (STS), has been deployed to manage the state explosion problem of SCDES. This paper aims to address this notorious issue of supervisory control with partial observation in the STS framework by state feedback control that calculates the controllers of the controllable-observable events only, which is realized by the following two steps. First, for a specification represented as a predicate, a supremal normal subpredicate that requires only the controllable-observable events enabled/disabled, is computed. Second, according to the new transition function constructed by the natural projection of the given STS, the supremal nonblocking, weakly controllable subpredicate is obtained from the supremal normal subpredicate. The proposed approach based on STS provides the possibility to supervise controllable events under partial observation in large-scale systems with the state explosion problem managed. An example with state size over 10^7 that leads to program crashes in SCDES can be solved in this paper. Moreover, in order to demonstrate the industrial applications of the contribution of this research, three examples are addressed.

1. Introduction

Discrete-event systems (DES) are discrete (in time and state space), asynchronous (event-driven rather than clock-driven), and in some sense generative (or nondeterministic). Supervisory control of DES (SCDES) [27–32,39,40], referred to as the Ramadge-Wonham (RW) framework developed in1980′s, provides a mathematical treatment for the modeling and control of DES. However, the state explosion is the main obstacle of SCDES attributed to the computational complexity of controllers. In order to break through this barrier, Ma and Wonham propose a hierarchical DES framework [20], namely *state-tree structures* (STS), equipped with dynamic and structured modules named *holons* (inspired by Harel [9]) and *state-trees*. Based on hierarchical state expansion relations, the study in [20] proposes an approach to model

☆ We appreciate Professor W. M. Wonham's support, creative discussion, and help in the past years to make this research happen. This work was supported in part by the Alexander von Humboldt Foundation, the National Natural Science Foundation of China under Grant nos. 61703322, 61374068, and 61673309, and the Science and Technology Development Fund, MSAR, under Grant No. 078/2015/A3.

* Corresponding author at: Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau.
E-mail addresses: guc@stu.xidian.edu.cn (C. Gu), wanggkingg@gmail.com (X. Wang), zhwli@xidian.edu.cn (Z. Li), nqwu@must.edu.mo (N. Wu).

https://doi.org/10.1016/j.ins.2018.07.025

DES hierarchically by STS, where the dynamics is described by holons. Naturally equipped with hierarchy (vertical structure) and concurrency (horizontal structure), the STS can be used to structure complex systems with the state size up to 10^{24}. However, if the computation is done on the basis of each state-tree, in the worst case, the computational complexity is as high as in SCDES. By analyzing the property of state-trees and *predicates*, the work in [11,19,20] reports a novel approach to the optimal nonblocking supervisory control design by defining *weak controllability* on predicates. As the basis of computation of the optimal supervisory control, predicates are represented by binary decision diagrams (BDD) [22]. A BDD is a rooted directed acyclic graph that is used to describe the controllers as a boolean function. With the BDD representation of predicates, the computational complexity of supervisor synthesis is no longer polynomial with respect to the model's state size, but to the number of BDD nodes in use [20]. Fortunately, in many practical cases, the size of BDD nodes is much smaller than that of states, which makes the computation of a supervisor with BDD efficient.

In SCDES, a DES is modeled by an automaton, and its behavior is described by the *formal languages* generated by the automaton over an alphabet of the event set. Observability is typically defined for the event observation: if sensors are located in dangerous environments or their signal collections and passing are difficult or expensive, users can define the corresponding events as unobservable. Considering the observability, the supervisory control and observation problem (SCOP) is investigated in [5,17,33], where a supervisor is constrained to observe the *controllable-observable* events (events are not only controllable but also observable) only. Furthermore, *normal* sublanguages [17], a relatively strict condition but positive to find the supremal sublanguage, are proposed to solve the SCOP, which means that only the controllable-observable events are allowed to be enabled/disabled. In the literature, many researchers also study this property extensively. Different approaches are investigated in [2,8,15,16,18,24,38] under the assumption that a part of the events in a system is observable. In [24], a necessary and sufficient condition is reported for the existence of a static-state feedback controller under partial observation. The research in [15] focuses on the problem of controlling DES by using predicates and predicate transformers. In addition, controlling DES under incomplete state observation is considered and observability of predicates is defined in [15]. Under controllability and observability constraints, the existence of a supervisory controller is certified. However, observability is a somewhat difficult property to deal with, and normality is such a restrictive condition that the supremal normal language is not maximally permissive. To fill the gap, the studies in [1,4,7,12,23,25,26,34–37] investigate different, necessary, and sufficient conditions for the supervisory control problem with partial observation, which are less restrictive than normality. Then synthesis algorithms are provided to construct safe and non-blocking supervisors, which might not be maximally permissive but is superior to the normal counterpart in yielding a generally larger controlled behavior. For most of these studies, after projecting out the unobservable events, a new generator is constructed where the states transiting through any unobservable path are combined. However, this process, known as the subset construction, dramatically increases the computational expense, which exacerbates the state explosion that remains to be tackled. Studies in [10,13,14] conduct excellent jobs in the reduction of the computational efforts of supervisory control problems with partial observation by dividing a complex system into several simple ones. In most cases, a coordinator is required to achieve nonblocking during the synthesis, which is a difficult problem. However, this can be easily achieved by the use of STS, which offers a compact representation of hierarchy and concurrency structure in finite state systems.

Since STS have the advantage over SCDES in managing the state explosion, it is worth to solve the supervisory control problem with partial observation in the STS framework. In this article, the problem with partial observation arising in STS models is solved based on predicates. First, the transition function is encoded as transformers between predicates, which is proved to be *sound* since it is equivalent to the counterparts of the original RW framework. It shows the reasonability of solving the problem with partial observation based on predicates while this problem is presented in view of the observability of events. In addition, thanks to BDD, predicates can be used in the computation of a supervisor, which reduces the computational overheads considerably. To solve the problem of STS with partial observation, the natural projection of predicates is defined first. Then, the controlled behavior can be obtained under a stringent constraint, *normality*, which requires that a normal predicate should be reached or reach others through observable events only. The key process consists of two steps: (1) given an STS and a supervisor represented by a predicate, find the supremal normal subpredicate; (2) considering the subpredicate as the new supervisor, and according to the new transition function that describes the system behavior via the observable event subset of the events only, compute the nonblocking and weakly controllable behavior. In addition, by noting that a system composed of 15 machines and 14 buffers causes the program crashes in SCDES, the controlled behavior under normality can be obtained under partial observation by the approach in this paper, which demonstrates the contribution of this research.

The rest of this paper is organized as follows. After the backgrounds on the RW framework and the STS model are introduced in Section 2, Section 3 elaborates upon the natural projection and exposes the projected components of the STS based on predicates. Section 4 investigates the normality such that supervisory control problems can be handled with a satisfactory solution. Section 5 dwells on three examples which are Small Factory, Control of a Guideway, and Cluster Tool. Section 6 discusses the advantages of using STS over the RW framework to analyze and solve the supervisory control problem with partial observation. Finally, Section 7 concludes this work.