Automatica 73 (2016) 8–14

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Online minimization of sensor activation for supervisory control☆

Weilin Wang

Department of Control Science and Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, PR China

ARTICLE INFO

Article history:
Received 4 June 2015
Received in revised form
2 December 2015
Accepted 30 May 2016

Keywords:
Discrete event systems
Online sensor activation
Supervisory control
Observability

ABSTRACT

In supervisory control, the objective of observation is to guarantee a correct control decision. To observe an event occurrence, an associated sensing device must be activated, which incurs a cost. In this paper, an online algorithm is developed to minimize sensor activation while ensuring that the collected information is sufficient. In previous work, it was determined that observation problems can be reduced to distinguishing certain pairs of states. Now, this is extended by taking into account future system evolution. Applying the extended result, the online algorithm only needs to look one step ahead and remember the current state estimate.

1. Introduction

The observation processes of control systems in the conventional model are usually cumbersome. In nature, even simple organisms possess the ability to perform fluid, dynamic, and self-regulating observation. By enabling the controller to flexibly activate / deactivate sensors according to the present situations, this work seeks to infuse an intuitive element of dynamic interplay into the observation of control systems.

In supervisory control, the control decisions depend on observability (Lin & Wonham, 1988; Tsitsiklis, 1989). Observability says that, following trajectories that appear to be identical, the controller must make the same decision. In the case of distributed control, different controllers need different observations, since each of them needs to implement different control actions. To accommodate this difference, observability was extended to coobservability for distributed control in Cieslak, Desclaux, Fawaz, and Varaiya (1988) and Rudie and Wonham (1992). Relevantly, the notion of diagnosability was invented for determining whether or not the information is sufficient for detecting and categorizing hidden faults within finite delay (Debouk, Lafortune, & Teneketzis, 2000; Jiang, Huang, Chandra, & Kumar, 2001; Jiang & Kumar, 2004; Lin, 1994; Moreira, Jesus, & Basilio, 2011; Qiu & Kumar, 2006; Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; Yoo & Lafortune, 2002).

In the conventional model, sensors for observing events are either always or never activated. Based on this assumption, optimization techniques are developed to compute the set of activated sensors with the minimum cost (Haji-Valizadeh & Loparo, 1996; Jiang, Kumar, & Garcia, 2003). This tactic is undesirable for a variety of reasons, including security, the lifespan of sensors, limited battery power, etc. When an agent is activating/deactivating sensors, the agent has different observations for different occurrences of the same event (Cassez & Tripakis, 2008; Shu, Huang, & Lin, 2013; Thorsley & Teneketzis, 2007; Wang, Lafortune, Girard, & Lin, 2010; Wang, Lafortune, Lin, & Girard, 2010). This is termed *dynamic observation* (Shu & Lin, 2010; Wang, Lafortune, & Lin, 2007; Wang, Girard, Lafortune, & Lin, 2011). Communication also causes dynamic observation (Ricker & Rudie, 1999; Wang, Lafortune, & Lin, 2008a,b).

The minimization of sensor activations was first studied for diagnosing the system (Cassez & Tripakis, 2008; Thorsley & Teneketzis, 2007). The intricate part is that the decision of activating a sensor at a given point in time depends on agent observation of the current system trajectory, which depends on how sensors have been activated in the past. This requirement was captured by information states in Thorsley and Teneketzis (2007). When the system is acyclic, the complexity of the approach is double exponential to the number of trajectories. Another approach is based on constructing most-permissive observers (Cassez & Tripakis, 2008; Dallal & Lafortune, 2014). The complexity for constructing such an observer is exponential to both the steps of the allowable delay, from a fault occurrence till the correct diagnosis being performed, and the size of the state space. In

☆ The material in this paper was partially presented at the 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference, December 16–18, 2009, Shanghai, China. This paper was recommended for publication in revised form by Associate Editor Jan Komenda under the direction of Editor Christos G. Cassandras.

E-mail address: wlwang@usst.edu.cn.

http://dx.doi.org/10.1016/j.automatica.2016.06.016

order to trade off between the quality of the solution and the amount of computation, the minimal sensor activation problem for diagnosability is also investigated using window-partition-based activation policies in Wang, Lafortune, Girard et al. (2010).

Recently, sensor activation is considered for the purpose of supervisory control in Wang, Lafortune, Lin et al. (2010). The agent activates sensors as needed in order to correctly implement the control law. With the solution space restricted to the transitions of the modeling automaton, polynomial algorithms that compute sets of minimal sensor activation policies are developed. In Wang, Lafortune, Lin et al. (2010), the calculation of minimal sensor activation policies is entirely off-line.

In this paper, an online algorithm is presented to minimize sensor activation in supervisory control. The hard constraint is that the agent must distinguish certain pairs of states, which is derived from a supervisory control law. To achieve fast online calculation, our online approach only requires the agent to look one step ahead. This counter-intuitive result is obtained using an off-line procedure to extend the specification by taking into account future evolution of the system. The forthcoming online algorithm only needs to estimate the current state, which achieves both "future independence" and "history independence" in online computation. To verify activation decisions, the agent only needs to compare state estimates to a pre-calculated extended specification immediately after observing a new event occurrence.

As compared to a policy computed by an offline approach (Wang, Lafortune, Girard et al., 2010; Wang, Lafortune, Lin et al., 2010), a distinctive advantage of the online computed policy is that its minimality is achieved on the domain of trajectories of the controlled system. Thus, solutions that are computed online are not subject to any further refinement. From an implementation standpoint, another distinctive feature of our online algorithm is that the computed policy is applied to control sensors on the fly, while offline computed policies require online implementation.

The rest of the paper is organized as follows. Section 2 starts with the definition of sensor activation policy, followed by a preliminary result on anti-monotonicity and a statement of the problem to be solved. Section 3 illustrates how to extend a specification to account for the future evolution of the system and studies properties of such an extension. Section 4 solves the problem using an online algorithm. Section 5 gives examples to illustrate the results. A preliminary and partial version of the result of this paper was previously presented without proofs (Wang, Lafortune, Lin, & Girard, 2009).

2. Problem formulation

2.1. Model of system with sensor activation

We model a discrete event system using a deterministic finite-state automaton $G = (Q, \Sigma, \delta, q_0)$, where Q is the finite set of states, Σ is the finite set of events, $\delta : Q \times \Sigma \to Q$ is the partial transition function where $\delta(q, \sigma) = q'$ means that there is a transition labeled by event σ from state q to state q', and q_0 is the initial state. δ is extended to $Q \times \Sigma^*$ in the usual way. $\mathcal{L}(G)$ is the language generated by G.[1]

Sensors are activated by an agent. The set of events that are potentially observable by the agent is denoted by Σ_o, and the set of events that are never observed is denoted by $\Sigma_{uo} = \Sigma \setminus \Sigma_o$. A sensor is associated with each observable event, it can be activated to make an occurrence of that event observable. If the sensor is not activated when the corresponding event occurs, the event occurrence is not observed.

When to activate sensors is described by the sensor activation mapping $\omega : \mathcal{L}(G) \to 2^{\Sigma_o}$. Specifically, $\omega(s)$, $s \in \mathcal{L}(G)$, is the subset of observable events corresponding to the set of activated sensors after s.

Given a sensor activation mapping ω, the corresponding information mapping $\theta^\omega : \mathcal{L}(G) \to \Sigma_o^*$ is inductively defined as follows. For the empty string ε, $\theta^\omega(\varepsilon) = \varepsilon$, and for all $s\sigma \in \mathcal{L}(G)$,

$$\theta^\omega(s\sigma) = \begin{cases} \theta^\omega(s)\sigma & \text{if } \sigma \in \omega(s) \\ \theta^\omega(s) & \text{otherwise.} \end{cases} \quad (1)$$

If σ occurs after s, it is observed iff the sensor for σ is activated when it occurs.

Notations for the system model and sensor activations:

- Given an s in a prefix-closed language L, let $PC(s) = \{u \in \Sigma^* : (\exists\ v \in \Sigma^*)\ uv = s\}$ be the prefix-closure of s. $|s|$, $s \in \mathcal{L}(G)$, is the number of event occurrences of s, called the length of s. s_n denotes the prefix of s with $|s_n| = n$, s_n is a system trajectory.
- Suppose sensor activation policies ω' and ω'' are defined over language L, then $\omega' \subseteq \omega''$ if $(\forall\ s \in L)\ \omega'(s) \subseteq \omega''(s)$. Moreover, $\omega' \subset \omega''$ if $\omega' \subseteq \omega''$ and $(\exists\ \ s \in L)\ \omega'(s) \subset \omega''(s)$. Similarly, for sensor activation policies ω, ω', and ω'' defined on language L, $\omega = \omega' \cup \omega''$ means, for all $s \in L$, $\omega(s) = \omega'(s) \cup \omega''(s)$.
- Suppose ω is an sensor activation policy defined on $\mathcal{L}(G)$, for a prefix-closed $L \subseteq \mathcal{L}(G)$, $\omega|_L$ denotes that ω is restricted to the smaller domain L. $\theta^{\omega|_L}$ is the information mapping defined on L with respect to $\omega|_L$.
- $P : \Sigma^* \to \Sigma_o^*$ is the natural projection.

2.2. Feasibility and anti-monotonicity

Sensor activation mapping ω must be consistent with the information mapping θ^ω that is built from it. To guarantee activation decisions to be practically feasible, any two strings that appear identical must be followed by the same sensor activation decision. Formally, ω is said to be *feasible* if

$$(\forall\ \sigma \in \Sigma)(\forall\ s\sigma, s'\sigma \in \mathcal{L}(G))\ \theta^\omega(s) = \theta^\omega(s')$$
$$\Rightarrow\ (\sigma \in \omega(s) \Leftrightarrow \sigma \in \omega(s')). \quad (2)$$

The agent updates its sensor activation decision immediately after it observes a new event occurrence.

In addition to (2), the agent must sufficiently activate sensors to distinguish state pairs included in a specification $T_{spec} \subseteq Q \times Q$. We assume that T_{spec} is given. How to obtain T_{spec} from control specifications is illustrated in Wang et al. (2007); Wang, Lafortune, Lin et al. (2010). Formally, with a given T_{spec}, it is required that no state pair $(q, q') \in T_{spec}$ is indistinguishable from the viewpoint of the agent, that is,

$$(\forall\ s, s' \in \mathcal{L}(G))\ \theta^\omega(s) = \theta^\omega(s')$$
$$\Rightarrow (\delta(q_0, s), \delta(q_0, s')) \notin T_{spec}. \quad (3)$$

We say that the policy ω satisfies T_{spec} if (3) holds.

We assume that, if all sensors are activated all the time, the specification T_{spec} can be satisfied, that is,

$$(\forall\ s, s' \in \mathcal{L}(G))\ P(s) = P(s')$$
$$\Rightarrow (\delta(q_0, s), \delta(q_0, s')) \notin T_{spec}. \quad (4)$$

To optimize sensor activation online, our strategy is to deactivate sensors one by one to see if the specification T_{spec} is violated. Checking whether a sensor can be deactivated or not requires the anti-monotonicity condition: the more an agent observes, the fewer state pairs it confuses. Theorem 1 for anti-monotonicity is from Wang, Lafortune, Lin et al. (2010).

Given ω, let $\mathcal{T}_{conf}(\omega) = \{(s, t) \in \mathcal{L}(G) \times \mathcal{L}(G) : \theta^\omega(s) = \theta^\omega(t)\}$ be the set of confusable string pairs.

[1] See Cassandras and Lafortune (2009) for standard procedures in discrete event systems.