Automatica 80 (2017) 162–171

Contents lists available at ScienceDirect

Automatica

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

A new approach for the verification of infinite-step and K-step opacity using two-way observers☆

Xiang Yin [a,1], Stéphane Lafortune [b]

[a] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

[b] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:
Received 13 June 2016
Received in revised form
3 November 2016
Accepted 27 January 2017

Keywords:
Discrete event systems
Infinite-step opacity
K-step opacity
Two-way observer

ABSTRACT

In the context of security analysis for information flow properties, where a potentially malicious observer (intruder) tracks the observed behavior of a given system, infinite-step opacity (respectively, K-step opacity) holds if the intruder can never determine for sure that the system was in a secret state for any instant within infinite steps (respectively, K steps) prior to that particular instant. We present new algorithms for the verification of the properties of infinite-step opacity and K-step opacity for partially-observed discrete event systems modeled as finite-state automata. Our new algorithms are based on a novel separation principle for state estimates that characterizes the information dependence in opacity verification problems, and they have lower computational complexity than previously-proposed ones in the literature. Specifically, we propose a new information structure, called the *two-way observer*, that is used for the verification of infinite-step and K-step opacity. Based on the two-way observer, a new upper bound for the delay in K-step opacity is derived, which also improves previously-known results.

1. Introduction

We investigate the verification of an important information-flow property called *opacity* that arises in security analysis of networked cyber and cyber–physical systems. We adopt a discrete-event framework, where the system under consideration is modeled as a partially-observed finite-state automaton and the security properties of interest for opacity are captured in terms of a set of secret states of the automaton. In this manner, the focus of the analysis is on the event-driven dynamics of the cyber or cyber–physical system of interest, captured in a discrete transition structure with unobservable events, and on the resulting observation properties during system operation. The system is said to be opaque if the secret cannot be revealed to an intruder that is potentially malicious. The intruder is modeled as an external observer that knows the transition structure of the system but can only observe part of the system's behavior.

To the best of our knowledge, the notion of opacity was initially introduced in Mazaré (2004), where it was motivated by the analysis of cryptographic protocols. It was then extended to the framework of Discrete Event Systems (DES) in Bryans, Koutny, Mazaré, and Ryan (2008) and Bryans, Koutny, and Ryan (2005). Several notions of opacity have been studied in order to capture different types of privacy requirements in the context of DES; among them we mention language-based opacity (Lin, 2011), current-state opacity (Saboori & Hadjicostis, 2007), initial-state opacity (Saboori & Hadjicostis, 2013), initial-and-final-state opacity (Wu & Lafortune, 2013), K-step opacity (Falcone & Marchand, 2014; Saboori & Hadjicostis, 2011b), and infinite-step opacity (Saboori & Hadjicostis, 2012b). If a given system is not opaque, then one is also interested in enforcing opacity. The opacity enforcement problem has been studied extensively under different enforcement mechanisms, e.g., using supervisory control (Badouel, Bednarczyk, Borzyszkowski, Caillaud, & Darondeau, 2007; Darondeau, Marchand, & Ricker, 2014; Dubreil, Darondeau, & Marchand, 2010; Saboori & Hadjicostis, 2012a; Takai & Kumar,

☆ This work was partially supported by NSF grants CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering) and CNS-1421122, and by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA. This work was done when the first author was at the University of Michigan. The material in this paper was presented at the 13th International Workshop on Discrete Event Systems, May 30–June 1, 2016, Xi'an, China. This paper was recommended for publication in revised form by Associate Editor Christoforos Hadjicostis under the direction of Editor Christos G. Cassandras.

E-mail addresses: xiangyin@umich.edu (X. Yin), stephane@umich.edu (S. Lafortune).

[1] Fax: +1 7347638041.

http://dx.doi.org/10.1016/j.automatica.2017.02.037
0005-1098/

2009; Takai & Oka, 2008; Yin & Lafortune, 2015b, 2016b), using dynamic observers (Cassez, Dubreil, & Marchand, 2012; Yin & Lafortune, 2015a; Zhang, Shu, & Lin, 2015), using insertion or edit functions (Wu & Lafortune, 2014; Wu, Raman, Lafortune, & Seshia, 2016), and using run-time techniques (Falcone & Marchand, 2014). Most of the above-mentioned works assume that the system is modeled as a finite-state automaton. Recently, the notion of opacity was extended to other classes of system models, including timed systems (Cassez, 2009), Petri nets (Bryans et al., 2005; Tong, Li, Seatzu, & Giua, 2016), pushdown systems (Kobayashi & Hiraishi, 2013), and stochastic systems (Bérard, Chatterjee, & Sznajder, 2015; Keroglou & Hadjicostis, 2013; Saboori & Hadjicostis, 2014). Several applications of opacity have also been investigated in the literature; see, e.g., Saboori and Hadjicostis (2011a) and Wu, Sankararaman, and Lafortune (2014). The reader is referred to the recent survey (Jacob, Lesage, & Faure, 2016) for more references on this active research area.

In this paper, we study the *verification* problem for the two notions of *infinite-step* opacity and *K-step* opacity. Current-state opacity requires that the secret not be revealed to the intruder based on the *current* state estimate. In contrast, infinite-step opacity requires that the secret not be revealed for any instant along the entire observation trajectory up to the present time, *based on the observations up to the current time*. Similarly, K-step requires that the secret not be revealed within K steps prior to the current instant, *based on the observations up to the current time*. It was shown in Wu and Lafortune (2013) that language-based opacity, initial-state opacity, and current-state opacity are "equivalent" in the sense that they can be mapped to one another in polynomial time. However, infinite-step and K-step opacity appear to be incomparable with the above notions, for the following reason. Whereas current-state opacity only depends on the current state estimate of the system, infinite-step and K-step opacity allow to do *smoothing*, i.e., to improve state estimation for *earlier* time instants, using observations up to the *present* time. Therefore, infinite-step and K-step opacity are fundamentally different from current-state opacity, language-based opacity, and initial-state opacity.

One of the motivations for studying infinite-step opacity and K-step opacity is that these two notions are very useful in privacy applications. For example, privacy is an important issue in Location-Based Services (LBS); see, e.g., Gruteser and Grunwald (2003). In LBS applications, the user may want to hide some of her crucial location information (e.g., visiting a bank or a hospital). However, this information may be revealed to an intruder located at the LBS server that keeps tracking the user's queries. Therefore, a formal methodology is needed in order to verify this privacy issue in LBS. It was shown in Wu et al. (2014) that verifying whether or not the user can always hide her *current* crucial location can be formulated as a current-state opacity verification problem. However, in some cases, the user may also want that the intruder never be able to infer that she was at a crucial place at some particular instant in the past (e.g., visited bank two days ago). Clearly, current-state opacity is not sufficient to capture this requirement, since the intruder may be able to use future observations to improve its knowledge about the user's location at some particular instant. However, this requirement can be captured using the notions of infinite-step or K-step opacity.

The notions of infinite-step opacity and K-step opacity were initially studied in Saboori and Hadjicostis (2011b, 2012b), respectively. More specifically, in Saboori and Hadjicostis (2011b), two different approaches for the verification of K-step opacity were proposed; both of these approaches have the same computational complexity of $O((|E_o|+1)^K \times |E_o| \times 2^{|X|})$, where X and E_o are the set of states and the set of observable events of the system, respectively. For infinite-step opacity, a verification algorithm of complexity of $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$ was provided in Saboori and Hadjicostis (2012b).

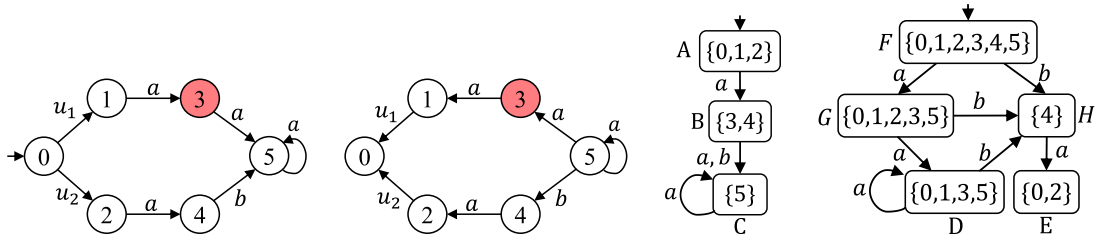
In this paper, we propose new approaches for the verification of infinite-step opacity and K-step opacity. Specifically, our contributions are summarized as follows.

- We provide a new characterization for the delayed state estimate, which is referred to as the *separation principle*. This result reveals that the information needed in the infinite-step (K-step) opacity verification problem can be decomposed into two mutually independent parts where each of them can be computed individually and effectively.
- We propose a novel information structure called the Two-Way Observer (TW-observer) in order to capture and represent in a single structure the two parts of independent information described by the separation principle.
- Based on the TW-observer, we present a new approach for the verification of infinite-step opacity. This approach results in a new algorithm that has complexity of $O(|E_o| \times 2^{|X|} \times 2^{|X|})$, compared with $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$ for the previous approach (Saboori & Hadjicostis, 2012b).
- We show that our proposed approach can also be used to verify the notion of K-step opacity, resulting in an algorithm of complexity of $O(\min\{2^{|X|}, |E_o|^K\} \times |E_o| \times 2^{|X|})$. This approach is based on the notion of K-reduced TW-observer that we introduce. The previous algorithm for verifying K-step opacity had a complexity of $O((|E_o|+1)^K \times |E_o| \times 2^{|X|})$ (Saboori & Hadjicostis, 2011b). Therefore, our new algorithm leads to considerable improvement in verification complexity when K is relatively large.
- Using the TW-observer, we provide a new upper bound in the K-step opacity problem. We show that a system is infinite-step opaque if and only if it is $(2^{|X|}-2)$-step opaque. This also improves upon the previous upper bound of $2^{|X|^2}-2$ derived in Saboori and Hadjicostis (2011b).
- Overall, the TW-observer provides a unified and more efficient framework for the verification of infinite-step and K-step opacity, as previous approaches require different techniques for verifying these properties.

In the definitions of infinite-step opacity and K-step opacity, it is required that the intruder cannot infer that the system was at a secret state for any *specific instant* in the past. However, in some cases, it is possible that the intruder knows that the system has visited a secret state in the past, although it cannot tell the specific instant (in terms of the number of steps) the secret state was visited. We call a system *trajectory-based* infinite-step (respectively, K-step) opaque if this scenario does *not* occur; examples for trajectory-based opacity can be found in Saboori and Hadjicostis (2011b, 2012b). Therefore, infinite-step (K-step) opacity is also referred to as *non-trajectory-based* infinite-step (K-step) opacity. Trajectory-based K-step opacity is referred to as K-step *strong* opacity in Falcone and Marchand (2014), where a verification algorithm is provided. Whether one needs to use the trajectory-based notions or the non-trajectory-based notions is application dependent. In this paper, we will focus on the non-trajectory-based notions.

The remaining sections of this paper are organized as follows. Sections 2 and 3 present the system model and the definitions of the opacity properties considered in this paper, respectively. In Section 4, the above-mentioned separation principle is investigated. Section 5 describes the structure of the proposed two-way observer and discusses its properties. Section 6 presents the new approach for the verification of infinite-step opacity. In Section 7, we show how to use the two-way observer to verify K-step opacity. Finally, we conclude the paper in Section 8.

Preliminary and partial versions of some of the results in this paper are presented, without proofs, in Yin and Lafortune (2016a).



(a) G. (b) G_R. (c) $Obs(G)$. (d) $Obs(G_R)$.

Fig. 1. System G with $E_o = \{a, b\}$ and $X_S = \{3\}$.

Herein, we provide all proofs that are omitted in Yin and Lafortune (2016a). In addition, we present new results that are not in Yin and Lafortune (2016a): (i) a detailed algorithm for the construction of the K-reduced TW-observer; and (ii) a new upper bound for K-step opacity.

2. System model

Let E be a finite set of events and E^* be the set of all finite strings over E including the empty string ϵ. A language $L \subseteq E^*$ is a subset of E^*. We denote by $\overline{L}$ the prefix-closure of L, i.e., $\overline{L} = \{u \in E^* : \exists v \in E^* \text{ s.t. } uv \in L\}$. For any string $s \in E^*$, $|s|$ denotes the length of t. We define $|\epsilon| = 0$.

A DES is modeled as a deterministic finite-state automaton

$$G = (X, E, f, X_0) \tag{1}$$

where X is the finite set of states, E is the finite set of events, $f : X \times E \to X$ is the deterministic (partial) transition function where $y = f(x, \sigma)$ means that there exists a transition labeled by event σ from state x to state y, and X_0 is the set of initial states. The transition function f is extended to domain $X \times E^*$ in the usual manner (see, e.g., Cassandras & Lafortune, 2008) and the extended function is still denoted by f. The language generated by G from state $x \in X$ is defined by $\mathcal{L}(G, x) = \{s \in E^* : f(x, s)!\}$, where ! means "is defined". For a set of states $Q \subseteq X$, we also define $\mathcal{L}(G, Q) = \cup_{x \in Q} \mathcal{L}(G, x)$. Therefore, the language generated by G is $\mathcal{L}(G) := \mathcal{L}(G, X_0)$. We assume that G is deterministic for the sake of simplicity, but the results developed hereafter can be easily extended to the case where G is nondeterministic. We also assume that G is accessible, i.e., states that are inaccessible from X_0 have been deleted.

Given $G = (X, E, f, X_0)$, we denote by $G_R = (X, E, f_R, X)$, the *reversed automaton* of G (Wu & Lafortune, 2013). Specifically, the transition function $f_R : X \times E \to 2^X$ is defined by: for any two states x and y in X and event $\sigma \in E$, we have $y = f(x, \sigma)$ iff $x \in f_R(y, \sigma)$. Note that G_R is *nondeterministic* in general. Also, the initial state of G_R is set to be the entire state space X, as a string in G can end at any state $x \in X$. Then, for any string $s = \sigma_1 \sigma_2 \ldots \sigma_{|s|} \in E^*$, we denote by s_R the reversed string of s, i.e., $s_R = \sigma_{|s|} \sigma_{|s|-1} \ldots \sigma_1$.

We assume that the intruder, which is modeled as an observer, has the full knowledge of the system's structure, but it can only partially observe the system's behavior. To this end, we assume that the event set E is partitioned into two disjoint subsets, E_o the set of observable events and E_{uo} the set of unobservable events, where $E_o \cup E_{uo} = E$ and $E_o \cap E_{uo} = \emptyset$. The natural projection $P : E^* \to E_o^*$ is defined by

$$P(\epsilon) = \epsilon \quad \text{and} \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in E_o \\ P(s) & \text{if } \sigma \in E_{uo}. \end{cases} \tag{2}$$

The natural projection is also extended to 2^{E^*}, i.e., for any $L \subseteq E^*$, $P(L) = \{t \in E_o^* : \exists s \in L \text{ s.t. } P(s) = t\}$.

Given a set of states $q \in 2^X$, we denote by $UR(q)$ the set of states that can be reached unobservably from some state in q, i.e.,

$$UR(q) := \{x \in X : \exists x' \in q, \exists s \in E_{uo}^* \text{ s.t. } f(x', s) = x\}.$$

We also denote by $Next(q, \sigma)$ the set of states that can be reached immediately upon the occurrence of observable event $\sigma \in E_o$, i.e.,

$$Next(q, \sigma) := \{x \in X : \exists x' \in q \text{ s.t. } f(x', \sigma) = x\}.$$

Then, the *observer* of G is defined by

$$Obs(G) = (Q_{obs}, E_o, f_{obs}, q_{obs,0}) \tag{3}$$

where $Q_{obs} \subseteq 2^X$, $q_{obs,0} = UR(X_0)$ and for any $q \in 2^X, \sigma \in E_o$, $f_{obs}(q, \sigma) = UR(Next(q, \sigma))$. In practice, we only build the accessible part of the observer, from its initial state $q_{obs,0}$.

We denote by $\hat{X}(s, G)$ the *current-state estimate* associated with observed string $s \in P(\mathcal{L}(G))$ w.r.t. G, i.e.,

$$\hat{X}(s, G) = \{x \in X : \exists x_0 \in X_0, \exists t \in \mathcal{L}(G, x_0) \text{ s.t. } f(x_0, t) = x \wedge P(t) = s\}.$$

In particular, for any string $s \in P(\mathcal{L}(G))$, we have that $f_{obs}(q_{obs,0}, s) = \hat{X}(s, G)$. Also, we denote by $Obs(G_R) = (Q_{obs,R}, E_o, f_{obs,R}, X)$ the observer of the reversed automaton G_R with initial state X.

Example 1. Consider the automaton G shown in Fig. 1(a), where $E_o = \{a, b\}$. The reversed automaton G_R of G is shown in Fig. 1(b), where all states are initial states. The observers $Obs(G)$ and $Obs(G_R)$ for automata G and G_R, are shown in Fig. 1(c) and (d), respectively. For example, for string $u_1 a \in \mathcal{L}(G)$, we have that $P(u_1 a) = a$ and $\hat{X}(a, G) = \{3, 4\} = f_{obs}(q_{obs,0}, a)$.

3. Opacity definitions

The system G has a set of secret states, denoted by $X_S \subseteq X$. We assume for simplicity that $X \setminus X_S$ is the set of non-secret states. We say that the system is *K-step opaque* if for any string that leads to a secret state, the intruder, which can observe the occurrences of events in E_o, can never determine for sure that the system is in a secret state at that point using up to K observations thereafter. We recall the formal definition from Saboori and Hadjicostis (2011b).

Definition 3.1 (*K-Step Opacity*)**.** Given system G, set of observable events E_o, set of secret states X_S, and non-negative integer $K \in \mathbb{N}$, system G is said to be K-step opaque (w.r.t. E_o and X_S) if

$$\begin{aligned} &(\forall x_0 \in X_0, \forall st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S \wedge |P(t)| \leq K) \\ &(\exists x_0' \in X_0, \exists s't' \in \mathcal{L}(G, x_0')) \\ &[f(x_0', s') \notin X_S \wedge P(s') = P(s) \wedge P(t') = P(t)]. \end{aligned} \tag{4}$$

When $K = 0$, K-step opacity reduces to current-state opacity. When $K \to \infty$, K-step opacity becomes infinite-step opacity. We recall the formal definition from Saboori and Hadjicostis (2012b).

Definition 3.2 (*Infinite-Step Opacity*)**.** Given system G, a set of observable events E_o, and a set of secret states X_S, system G is said to be infinite-step opaque (w.r.t. E_o and X_S) if

$$(\forall x_0 \in X_0, \forall st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S)$$
$$(\exists x_0' \in X_0, \exists s't' \in \mathcal{L}(G, x_0'))$$
$$[f(x_0', s') \notin X_S \wedge P(s') = P(s) \wedge P(t') = P(t)]. \tag{5}$$

Example 2. Consider again the system G in Fig. 1(a). Let $X_S = \{3\}$ be the set of secret states. It is easy to verify that Eq. (4) does not hold for $K = 1$. By taking $s = u_1 a$ and $t = a$, we know that the only string $s't' \in \mathcal{L}(G)$ such that $P(s') = P(s)$ and $P(t') = P(t)$ is st itself. Intuitively, this means that after observing aa, the intruder will know for sure that the system was in secret state 3 one step earlier. Therefore, G is not 1-step opaque w.r.t. E_o and X_S, which also implies that G is not infinite-step opaque. However, this system is current-state opaque (or 0-step opaque), since the intruder can never determine whether or not the system is currently in a secret state.

In Saboori and Hadjicostis (2011b, 2012b), different approaches for the verification of K-step opacity and infinite-step opacity are provided. Specifically, in Saboori and Hadjicostis (2011b), two approaches called the state mapping-based approach and the observation sequence-based approach are provided for the verification of K-step opacity; both of them have (worst-case) time complexity of $O(|E_o| \times (|E_o|+1)^K \times 2^{|X|})$.[2] In Saboori and Hadjicostis (2012b), an algorithm for the verification of infinite-step opacity is proposed by using a bank of initial-state estimators, which has (worst-case) time complexity of $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$. The reader is referred to Saboori and Hadjicostis (2011b, 2012b) for more details. Hereafter, we will provide a uniform and more efficient approach for the verification of K-step and infinite-step opacity.

4. Delayed state estimate and its characterization

In this section, we first show how infinite-step opacity can be characterized by using the delayed state estimate that was originally proposed in order to characterize K-step opacity (Saboori & Hadjicostis, 2011b). Then, we provide a *separation principle for the delayed state estimate*[3] by dividing it into two independent components.

First, we recall the definition of delayed state estimate from Saboori and Hadjicostis (2011b).

Definition 4.1. Let $s = \sigma_1\sigma_2\ldots\sigma_n \in P(\mathcal{L}(G))$. Let $K \leq n$ be a non-negative integer. Then, the K-delayed state estimate associated with s, denoted by $\hat{X}_{|s|-K}(s)$, is defined as the set of states the system could have been in K steps earlier, after observing s. Mathematically, we have

$$\hat{X}_{|s|-K}(s) := \{x \in X : \exists x_0 \in X_0, \exists t_1 t_2 \in \mathcal{L}(G, x_0) \text{ s.t. } x = f(x_0, t_1) \wedge P(t_1) = \sigma_1\sigma_2\ldots\sigma_{n-K} \wedge P(t_2) = \sigma_{n-K+1}\ldots\sigma_n\}.$$

Clearly, the delayed estimate is a generalization of both the initial-state estimate and the current-state estimate. For any string $s \in P(\mathcal{L}(G))$, $\hat{X}_{|s|-K}(s)$ becomes the initial-state estimate when $K = |s|$ and becomes the current-state estimate when $K = 0$. Note that $\hat{X}_{|s|-K}(s)$ is always a non-empty set for any $s \in P(\mathcal{L}(G))$, $K \leq |s|$.

It was shown in Saboori and Hadjicostis (2011b) that the system G is K-step opaque, if and only if,

$$\forall s \in P(\mathcal{L}(G)), \forall k \leq \min\{K, |s|\} : \hat{X}_{|s|-k}(s) \not\subseteq X_S. \tag{6}$$

Similarly, the next result says that infinite-step opacity can be characterized by the delayed state estimate, if we do not set the delay to a fixed K. For this purpose, we define

$$\hat{X}_{|s|}(st) := \{x \in X : \exists x_0 \in X_0, \exists t_1 t_2 \in \mathcal{L}(G, x_0) \text{ s.t. } x = f(x_0, t_1) \wedge P(t_1) = s \wedge P(t_2) = t\}.$$

Proposition 1. *The system G is infinite-step opaque (w.r.t. X_S and E_o) if and only if*

$$\forall st \in P(\mathcal{L}(G)) : \hat{X}_{|s|}(st) \not\subseteq X_S. \tag{7}$$

Proof. ($\Rightarrow$) By contrapositive. Suppose that there exists a string $st \in P(\mathcal{L}(G))$ such that $\hat{X}_{|s|}(st) \subseteq X_S$. Then, $\hat{X}_{|s|}(st) \subseteq X_S$ implies that $\forall x_0 \in X_0, \forall uv \in \mathcal{L}(G, x_0)$ such that $P(u) = s$ and $P(v) = t$, we have that $f(x_0, u) \in X_S$. Therefore, taking strings u and v above as the strings s and t in Eq. (5), respectively, we know that Eq. (5) does not hold, i.e., G is not infinite-step opaque.

($\Leftarrow$) By contradiction. Suppose that Eq. (7) holds and assume that G is not infinite-step opaque. Since Eq. (5) does not hold, we know that

$$(\exists x_0 \in X_0, \exists st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S)$$
$$(\forall x_0' \in X_0, \forall s't' \in \mathcal{L}(G, x_0'))$$
$$[P(s') = P(s) \wedge P(t') = P(t) \Rightarrow f(x_0', s') \in X_S]. \tag{8}$$

However, by Eq. (7), we know that for any string $uv \in P(\mathcal{L}(G))$, $\exists x_0'' \in X_0, \exists s''t'' \in \mathcal{L}(G, x_0'')$ such that $P(s'') = u, P(t'') = v$ and $f(x_0'', s'') \notin X_S$. This contradicts Eq. (8). □

Observe that for any $st \in P(\mathcal{L}(G))$, the delayed state estimate $\hat{X}_{|s|}(st)$ can never by empty. Computing $\hat{X}_{|s|}(st)$ for a string $st \in P(\mathcal{L}(G))$ is not a easy task, since it not only depends on the information available at the point when s is observed, but also depends on the additional information obtained thereafter from suffix t. Moreover, the length of the suffix t can be unbounded in general. This is also the essential difference between infinite-step opacity and current/initial-state opacity.

Next, we present one of the key results in this paper, which is also referred to as the *separation principle* hereafter. It reveals that for any string $st \in P(\mathcal{L}(G))$, the delayed state estimate $\hat{X}_{|s|}(st)$ consists of two parts that only depend on string s and string t, respectively.

Theorem 2. *For any string $st \in P(\mathcal{L}(G))$, we have that*

$$\hat{X}_{|s|}(st) = \hat{X}(s, G) \cap \hat{X}(t_R, G_R) \tag{9}$$

or, equivalently,

$$\hat{X}_{|s|}(st) = f_{obs}(q_{obs,0}, s) \cap f_{obs,R}(X, t_R). \tag{10}$$

Proof. First, we recall from Wu and Lafortune (2013) that the reversed automaton and the reversed strings satisfy the following facts:

[2] This complexity was originally expressed as $O((|E_o|+1)^K \times 2^{|X|})$ in Saboori and Hadjicostis (2011b) because it only considers the number of states in the state estimator structure. In order to obtain the time complexity, the original complexity should be multiplied by $|E_o|$, namely, we also need to consider the number of transitions in the structure.

[3] There are other unrelated separation principles in linear system theory (Chen, 1995) and optimal stochastic control (Varaiya & Kumar, 1986).

Fact1: $x' \in f_R(x, t_R) \Leftrightarrow x = f(x', t)$
Fact2: $P(t_R) = P(t'_R) \Leftrightarrow P(t) = P(t')$.

Now, we are ready to show that $\hat{X}_{|s|}(st) = f_{obs}(q_{obs,0}, s) \cap f_{obs,R}(X, t_R)$.

$$
\begin{aligned}
& x \in \hat{X}_{|s|}(st) \\
\Leftrightarrow & \exists x_0 \in X_0, \exists s't' \in \mathcal{L}(G, x_0) : \\
& x = f(x_0, s') \wedge P(s') = s \wedge P(t') = t \\
\Leftrightarrow & [\exists x_0 \in X_0, \exists s' \in \mathcal{L}(G, x_0) : P(s') = s \wedge f(x_0, s') = x] \wedge \\
& [\exists x' \in X, \exists t' \in \mathcal{L}(G, x) : P(t') = t \wedge f(x, t') = x'] \\
\Leftrightarrow & [\exists x_0 \in X_0, \exists s' \in \mathcal{L}(G, x_0) : P(s') = s \wedge f(x_0, s') = x] \wedge \\
& [\exists x' \in X, \exists t'_R \in \mathcal{L}(G_R, x') : P(t'_R) = t_R \wedge x \in f_R(x', t'_R)] \\
\Leftrightarrow & x \in f_{obs}(X_0, s) \wedge x \in f_{obs,R}(X, t_R) \\
\Leftrightarrow & x \in f_{obs}(q_{obs,0}, s) \cap f_{obs,R}(X, t_R).
\end{aligned}
$$

Note that the third equivalence follows from Facts 1 and 2 above. □

We illustrate the above result by the following example.

Example 3. Let us go back to Example 2. Consider strings $s = a$ and $t = a$ such that $st \in P(\mathcal{L}(G))$. We have $t_R = t = a$. Then, according to Theorem 2, we know that

$$
\begin{aligned}
\hat{X}_{|s|}(st) &= f_{obs}(q_{obs,0}, a) \cap f_{obs,R}(X, a) \\
&= \{3, 4\} \cap \{0, 1, 2, 3, 5\} \\
&= \{3\} \subseteq X_S.
\end{aligned}
$$

Therefore, by Proposition 1, we know that G is not infinite-step opaque w.r.t. E_o and X_S.

Theorem 2 has the following important implications. It reveals that given a string s and its suffix t, the delayed state estimate $\hat{X}_{|s|}(st)$ essentially consists of two parts of information: the pre-information obtained by observing s, i.e., $f_{obs}(X_0, s)$ and the post-information obtained thereafter by observing t, i.e., $f_{obs,R}(X, t_R)$. More importantly, these two information sets are mutually independent or *separated*, i.e., $\hat{X}_{|s|}(st)$ can be calculated by simply taking the intersection of the pre-information with the post-information. In other words, computing the post-information does not depend on where the suffix t comes from. It can be simply calculated by using the reversed automaton from initial state X, i.e., we assume that there is no pre-knowledge about where t comes from, since this information will be "taken care of" by $f_{obs}(X_0, s)$. However, $P(\mathcal{L}(G))$ contains an infinite number of strings in general, and for each string in $P(\mathcal{L}(G))$, we need to know at what point we should divide it into its pre-information and its post-information. In other words, we need to build a finite structure in order to capture all strings in $P(\mathcal{L}(G))$ and all possible breakpoints for each string. This idea is formalized by the structure of "two-way observer", which is defined in the next section.

5. Two-way observer

In this section, we define the notion of *Two-Way Observer* (TW-Observer), which essentially asynchronously composes the observer of G and the observer of G_R. Then, we discuss the properties of the TW-observer.

We start by defining the TW-observer.

Definition 5.1. The *Two-Way Observer* of G is a deterministic finite-state automaton

$$Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0}) \tag{11}$$

where

- $Q_{TW} \subseteq Q_{obs} \times Q_{obs,R}$ is the set of states;
- $E_{TW} = (E_o \times \{\epsilon\}) \cup (\{\epsilon\} \times E_o)$ is the set of events;
- $q_{TW,0} = (q_{obs,0}, X)$ is the initial state;
- $f_{TW} : Q_{TW} \times E_{TW} \to Q_{TW}$ is the (deterministic) transition function defined by: for any state $(q_1, q_2) \in Q_{TW}$ and event $\sigma \in E_o$, the following transitions are defined whenever they are feasible

$$f_{TW}((q_1, q_2), (\sigma, \epsilon)) = (f_{obs}(q_1, \sigma), q_2) \tag{12}$$
$$f_{TW}((q_1, q_2), (\epsilon, \sigma)) = (q_1, f_{obs,R}(q_2, \sigma)). \tag{13}$$

As for the other automata in this paper, we only consider the accessible part of $Obs_{TW}(G)$ from its initial state.

Intuitively, the TW-observer tracks a string s in $P(\mathcal{L}(G))$ from $q_{obs,0}$ and a reversed string t_R in $Rev(P(\mathcal{L}(G)))$ from X, where $Rev(L) = \{s_R : s \in L\}$. Let (q_1, q_2) be a state reached in $Obs_{TW}(G)$. If $q_1 \cap q_2 \neq \emptyset$, then it means that the above two strings s and t_R "coincide" at some state. In other words, this state could be a "breakpoint" for some string in $\mathcal{L}(G)$, since some strings s' and t' such that $P(s') = s$ and $P(t') = t$ can be "connected" at a state in $q_1 \cap q_2$, i.e., $s't' \in \mathcal{L}(G)$.

Before we formalize the above discussion, we introduce some necessary notions. For any string $t \in \mathcal{L}(Obs_{TW}(G))$, we denote by $\tau_1(t) \in E_o^*$ and $\tau_2(t) \in E_o^*$ the first and second components of string t, respectively. For example, if $t = (a, \epsilon)(a, \epsilon)(\epsilon, b)$, then $\tau_1(t) = aa$ and $\tau_2(t) = b$.

Lemma 3. *Let $t \in \mathcal{L}(Obs_{TW}(G))$ be a string in the* TW*-observer and $f_{TW}(q_{TW,0}, t) = (q_1, q_2)$ be the state reached by t. Then, we have*

$$q_1 \cap q_2 \neq \emptyset \Rightarrow (\exists s \in \mathcal{L}(G))[\tau_1(t)(\tau_2(t))_R = P(s)] \tag{14}$$

or, equivalently,

$$q_1 \cap q_2 \neq \emptyset \Rightarrow \tau_1(t)(\tau_2(t))_R \in P(\mathcal{L}(G)). \tag{15}$$

Proof. By the construction of $Obs_{TW}(G)$, we know that $q_1 = f_{obs}(X_0, \tau_1(t))$ and $q_2 = f_{obs,R}(X, \tau_2(t))$. Let $x \in q_1 \cap q_2$ be a state in X. Then, we know that

$$\exists x_0 \in X_0, \exists s_1 \in \mathcal{L}(G, x_0) : P(s_1) = \tau_1(t) \wedge f(x_0, s_1) = x \tag{16}$$

and

$$\exists x' \in X, \exists s_2 \in \mathcal{L}(G_R, x') : P(s_2) = \tau_2(t) \wedge x \in f_R(x', s_2). \tag{17}$$

However, $x \in f_R(x', s_2)$ implies that $f(x, (s_2)_R) = x'$ and $P(s_2) = \tau_2(t)$ implies that $P((s_2)_R) = (\tau_2(t))_R$. Therefore, we know that exists a string $s = s_1(s_2)_R \in \mathcal{L}(G)$ such that $P(s) = P(s_1)P((s_2)_R) = \tau_1(t)(\tau_2(t))_R$. □

Similarly, for any string $s_1 s_2 \in \mathcal{L}(G)$, we can find a corresponding string $t \in \mathcal{L}(Obs_{TW}(G))$ such that the first component of t is $P(s_1)$ and the second component of t is the reversed string of $P(s_2)$. This is formalized by the following lemma.

Lemma 4. *For any string $s = s_1 s_2 \in P(\mathcal{L}(G))$, there exists a string $t \in \mathcal{L}(Obs_{TW}(G))$ such that $\tau_1(t) = s_1$ and $(\tau_2(t))_R = s_2$.*

Proof. Let $s_1 = \sigma_1^1 \sigma_2^1 \ldots \sigma_{|s_1|}^1$ and $s_2 = \sigma_1^2 \sigma_2^2 \ldots \sigma_{|s_2|}^2$. Consider the following string

$$
\begin{aligned}
t = & (\sigma_1^1, \epsilon)(\sigma_2^1, \epsilon) \cdots (\sigma_{|s_1|}^1, \epsilon) \\
& (\epsilon, \sigma_{|s_2|}^2)(\epsilon, \sigma_{|s_2|-1}^2) \cdots (\epsilon, \sigma_1^2).
\end{aligned} \tag{18}
$$

Since $s_1 s_2 \in P(\mathcal{L}(G))$, we know that $(s_2)_R \in P(\mathcal{L}(G_R, X))$. Therefore, we have that $s_1 \in \mathcal{L}(Obs(G))$ and $(s_2)_R \in \mathcal{L}(Obs(G_R))$ Then, by the construction of $Obs_{TW}(G)$, we know that $t \in \mathcal{L}(Obs_{TW}(G))$. Moreover, we have that $\tau_1(t) = s_1$ and $(\tau_2(t))_R = s_2$ by the construction of t. □

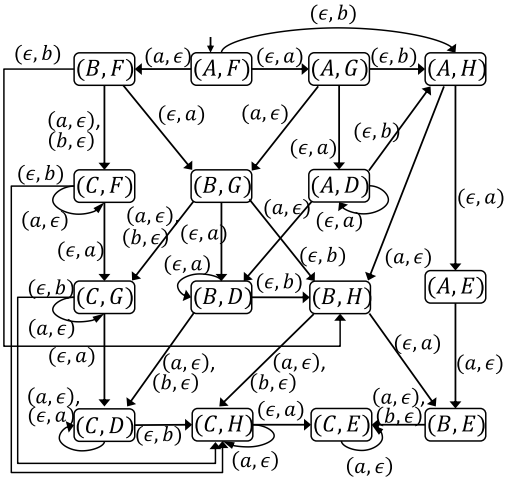


Fig. 2. The two-way observer $Obs_{TW}(G)$ for the system in Fig. 1(a).

Note that the string t constructed in the above proof is not unique in general. For example, we could have taken

$$t' = (\epsilon, \sigma^2_{|s_2|})(\epsilon, \sigma^2_{|s_2|-1}) \cdots (\epsilon, \sigma^2_1) \\ (\sigma^1_1, \epsilon)(\sigma^1_2, \epsilon) \cdots (\sigma^1_{|s_1|}, \epsilon) \tag{19}$$

and we would still have obtained $t' \in \mathcal{L}(Obs_{TW}(G))$.

The next example illustrates the TW-observer and its properties.

Example 4. Again, we consider the system G in Fig. 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. The TW-observer $Obs_{TW}(G)$ for this system is shown in Fig. 2. For the sake of simplicity, for each state in Q_{TW}, the first and the second components of the state are depicted by using short-hand notation according to Fig. 1(c) and (d), respectively. For example, state (C, D) represents state $(\{5\}, \{0, 1, 3, 5\})$, which can be reached by string $(a, \epsilon)(b, \epsilon)(\epsilon, a)(\epsilon, a)$. Since $\{5\} \cap \{0, 1, 3, 5\} = \{5\} \neq \emptyset$, by Lemma 3, we know that string $ab(aa)_R = abaa$ exists in $P(\mathcal{L}(G))$.

Remark 1. Note that the TW-observer in this paper is defined for deterministic finite-state automata. As we mentioned earlier, it can also be extended to the nondeterministic case, where the transition function is in the form of $f : X \times E \to 2^X$. Observe that we want each string $(s_1, s_2) \in E^*_{TW}$ to lead to state $(\hat{X}(s_1, G), \hat{X}(s_2, G_R))$ in the TW-observer. In the nondeterministic setting, the state estimate $\hat{X}(s, G)$ simply becomes $\hat{X}(s, G) = \{x \in X : \exists x_0 \in X_0, \exists t \in \mathcal{L}(G, x_0) \text{ s.t. } x \in f(x_0, t) \wedge P(t) = s\}$. It still can be computed recursively by constructing the observer automaton, which has the same complexity as the deterministic case; see, e.g., Cassandras and Lafortune (2008). Therefore, the TW-observer for a nondeterministic automaton also has at most $2^{2|X|}$ states in the worst case. This result is consistent with the fact that nondeterminism does not introduce additional complexity for the purpose of *verification* in *partially-observed* DES.

6. Verification of infinite-step opacity

In this section, we use the results developed so far and propose a new algorithm for the verification of infinite-step opacity.

According to Theorem 2, we see that the states in the TW-observer essentially capture all possible combinations of $\hat{X}(s, G)$ and $\hat{X}(t_R, G_R)$. Therefore, if the system is infinite-step opaque, then there should not exist a state in $Obs_{TW}(G)$ such that the intersection of its first and second components is a subset of secret states. This idea is formalized by the following theorem, which reveals that, in order to verify infinite-step opacity, it suffices to check whether or not the TW-observer contains a state in which the intersection of the two components is a subset of the set of secret states.

Theorem 5. *Let G be the system automaton, E_o be the set of observable events, and X_S be the set of secret states. Let $Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0})$ be the* TW-Observer *of G. Then, G is infinite-step opaque w.r.t. E_o and X_S if and only if*

$$\forall (q_1, q_2) \in Q_{TW} : q_1 \cap q_2 \not\subseteq X_S \text{ or } q_1 \cap q_2 = \emptyset. \tag{20}$$

Proof. ($\Rightarrow$) By contraposition. Suppose that there exists a state $(q_1, q_2) \in Q_{TW}$ such that $q_1 \cap q_2 \subseteq X_S$ and $q_1 \cap q_2 \neq \emptyset$. Let $s \in \mathcal{L}(Obs_{TW}(G))$ be a string that leads to this state, i.e., $f_{TW}(q_{TW,0}, s) = (q_1, q_2)$. Since $q_1 \cap q_2 \neq \emptyset$, by Lemma 3, we know that

$$\tau_1(s)(\tau_2(s))_R \in P(\mathcal{L}(G)).$$

By the construction of $Obs_{TW}(G)$, we have that $q_1 = f_{obs}(X_0, \tau_1(s))$ and $q_2 = f_{obs,R}(X, \tau_2(s))$. Therefore, by Theorem 2, we know that for string $\tau_1(s)(\tau_2(s))_R$, we have that

$$\begin{aligned} &\hat{X}_{|\tau_1(s)|}(\tau_1(s)(\tau_2(s))_R) \\ &\quad = f_{obs}(X_0, \tau_1(s)) \cap f_{obs,R}(X, ((\tau_2(s)_R))_R) \\ &\quad = f_{obs}(X_0, \tau_1(s)) \cap f_{obs,R}(X, \tau_2(s)) \\ &\quad = q_1 \cap q_2 \\ &\quad \subseteq X_S. \end{aligned}$$

Therefore, by Proposition 1, G is not infinite-step opaque w.r.t. E_o and X_S.

($\Leftarrow$) Also by contraposition. Suppose that G is not infinite-step opaque w.r.t. E_o and X_S, which means that there exists a string $s_1 s_2 \in P(\mathcal{L}(G))$, such that $\hat{X}_{|s_1|}(s_1 s_2) \subseteq X_S$. By Lemma 4, we know that there exists a string $t \in \mathcal{L}(Obs_{TW}(G))$ such that $q_1 \cap q_2 \neq \emptyset$ and $\tau_1(t) = s_1$ and $(\tau_2(t))_R = s_2$, where $f_{TW}(q_{TW,0}, t) = (f_{obs}(X_0, \tau_1(t)), f_{obs,R}(X, \tau_2(t))) =: (q_1, q_2)$. Since

$$\hat{X}_{|s_1|}(s_1 s_2) = f_{obs}(X_0, s_1) \cap f_{obs,R}(X, (s_2)_R)$$

we know that

$$\hat{X}_{|s_1|}(s_1 s_2) = q_1 \cap q_2 \subseteq X_S.$$

Moreover, since $s_1 s_2 \in P(\mathcal{L}(G))$, we know that $\hat{X}_{|s_1|}(s_1 s_2) = q_1 \cap q_2 \neq \emptyset$. Overall, we know that Eq. (20) does not hold. □

Remark 2. When the system is not infinite-step opaque, based on the "$\Rightarrow$" direction in the proof of Theorem 5, we can find a string $\tau_1(s)(\tau_2(s))_R \in P(\mathcal{L}(G))$ such that the intruder knows for sure that the system was in a secret state $|\tau_2(s)|$-steps earlier when $\tau_1(s)(\tau_2(s))_R$ is observed. In other words, string $\tau_1(s)(\tau_2(s))_R$ can be provided to the user as a cause of the violation of infinite-step opacity when the verification result is negative.

The following example illustrates how to use the above theorem for the verification of infinite-step opacity.

Example 5. We still consider the system G in Fig. 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. The TW-observer $Obs_{TW}(G)$ is shown in Fig. 2. We see that state (B, G), which denotes state $(\{3, 4\}, \{0, 1, 2, 3, 5\})$, is reached by string $(a, \epsilon)(\epsilon, a)$ or string $(\epsilon, a)(a, \epsilon)$. Since $\{3, 4\} \cap \{0, 1, 2, 3, 5\} = \{3\} \subseteq X_S$, by Theorem 5, we know that G is not infinite-step opaque.

Remark 3. We discuss the time complexity of the above approach for the verification of infinite-step opacity. Clearly, in the worst case, there are at most $2^{|X|} \times 2^{|X|}$ states and $|E_o| \times 2^{|X|} \times 2^{|X|}$ transitions in the TW-observer. Therefore, the (worst-case) time complexity of the proposed algorithm is of $O(|E_o| \times 2^{|X|} \times 2^{|X|})$.

Notice that the complexity of this TW-observer-based verification algorithm is smaller than that of the existing algorithm proposed in Saboori and Hadjicostis (2012b), which is of $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$, as was mentioned earlier. It was shown in Saboori and Hadjicostis (2012b) that the verification of infinite-step opacity is PSPACE-hard. Therefore, it seems highly unlikely that there exists a polynomial-time algorithm for the verification of infinite-step opacity.

7. Verification of K-step opacity

In this section, we discuss the verification of K-step opacity. First, we propose an approach by using the TW-observer directly. Then we provide a more efficient approach by using the notion of *K-reduced TW-observer* that we define. Finally, we provide a new upper bound for the delay, which improves upon previously-known results.

7.1. Verifying K-step opacity using the TW-observer

The following theorem shows how the TW-observer can be used directly to verify K-step opacity.

Theorem 6. *Let G be the system automaton, E_o be the set of observable events, and X_S be the set of secret states. Let $Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0})$ be the* TW-*Observer of G. Then, G is K-step opaque w.r.t. E_o and X_S, if and only if, for any string $s \in \mathcal{L}(Obs_{TW}(G))$ such that $f_{TW}(q_{TW,0}, s) = (q_1, q_2)$, we have that*

$$[q_1 \cap q_2 \subseteq X_S \wedge q_1 \cap q_2 \neq \emptyset] \Rightarrow |\tau_2(s)| > K. \tag{21}$$

Proof. ($\Rightarrow$) By contrapositive. Suppose that there exists a string $s \in \mathcal{L}(Obs_{TW}(G))$ such that $f_{TW}(q_{TW,0}, s) = (q_1, q_2)$, $q_1 \cap q_2 \subseteq X_S$, $q_1 \cap q_2 \neq \emptyset$, and $|\tau_2(s)| \leq K$. Since $q_1 \cap q_2 \neq \emptyset$, by Lemma 3, we know that $\tau_1(s)(\tau_2(s))_R \in P(\mathcal{L}(G))$. Moreover, since

$$q_1 \cap q_2 = \hat{X}_{|\tau_1(s)(\tau_2(s))_R|-|(\tau_2(s))_R|}(\tau_1(s)(\tau_2(s))_R) \subseteq X_S$$

and $|(\tau_2(s))_R| = |\tau_2(s)| \leq K$, we know that G is not K-step opaque since Eq. (6) is violated.

($\Leftarrow$) Also by contraposition. Suppose that G is not K-step opaque, which means that $\exists s_1 s_2 \in P(\mathcal{L}(G))$ such that $\hat{X}_{|s_1 s_2|-|s_2|}(s_1 s_2) \subseteq X_S$ and $|s_2| \leq K$. By Lemma 4, we know that there exists a string $s \in \mathcal{L}(Obs_{TW}(G))$ such that $\tau_1(s) = s_1$ and $\tau_2(s) = (s_2)_R$. Let $f_{TW}(q_{TW,0}, s) = (q_1, q_2)$. Then, we know that

$$\begin{aligned} q_1 \cap q_2 &= f_{obs}(q_{obs,0}, s_1) \cap f_{obs,R}(X, (s_2)_R) \\ &= \hat{X}_{|s_1 s_2|-|s_2|}(s_1 s_2) \\ &\subseteq X_S \end{aligned}$$

and $|\tau_2(s)| = |s_2| \leq K$. Moreover, $q_1 \cap q_2 \neq \emptyset$, since $\hat{X}_{|s_1 s_2|-|s_2|}(s_1 s_2)$ is always non-empty. This completes the contrapositive proof. □

Theorem 6 immediately suggests an approach to verify K-step opacity. First, we construct a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where each vertex in $\mathcal{V}$ corresponds to a state in $Obs_{TW}(G)$, each edge in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ corresponds to a transition in $Obs_{TW}(G)$ and the weight function $w : \mathcal{E} \to \{0, 1\}$ assigns each edge a zero weight if its corresponding event is of the form (σ, ϵ) and a unit weight if its corresponding event is of the form (ϵ, σ). Then we compute the minimum weight of paths from the initial vertex to a vertex which corresponds to state (q_1, q_2) such that $q_1 \cap q_2 \neq \emptyset$ and $q_1 \cap q_2 \subseteq X_S$. If the minimum weight computed is larger than K, then we know that G is K-step opaque. Note that finding the minimum weight can be done in $O(|\mathcal{V}| + |\mathcal{E}|)$. Hence, the (worst-case) time complexity of this approach is of $O(|E_o| \times 2^{|X|} \times 2^{|X|})$.

7.2. The K-reduced TW-observer

In the approach for verifying K-step opacity presented in Section 7.1, the complexity of $O(|E_o| \times 2^{|X|} \times 2^{|X|})$ comes from the construction of the *entire* TW-observer. However, by exploiting Theorem 6, we can reduce this worst-case complexity, since there is no need to construct the entire TW-observer or the corresponding graph. Namely, it suffices to check whether or not we can reach a "secret-revealing" state (i.e., a state (q_1, q_2) such that $\emptyset \neq q_1 \cap q_2 \subseteq X_S$) from the initial state *within K edges of unit value*. In other words, to verify K-step opacity, it suffices to construct part of the TW-observer structure, in which all states can be reached from the initial state via some string whose length of the second component is smaller than or equal to K. We call this structure the *K-reduced TW-observer*, and denote it by $Obs^K_{TW} = (Q^K_{TW}, E_{TW}, f^K_{TW}, q^K_{TW,0})$.

Formally, Obs^K_{TW} is constructed by Algorithm 1, which works as follows. First, we perform a K-step breadth-first search (procedure BFS) from the initial state $(q_{obs,0}, X)$, the same initial state as the TW-observer. In this procedure, integer D is used to denote the shortest distance of the second component between a state and the initial state. Therefore, we only consider $D \leq K$. For each D, $\Xi(D)$ denotes the set of states whose distances from the initial state are D. Note that, the first component of each state reached in this step stays at the initial state $q_{obs,0}$, since all transitions introduced in this step are in the form of (ϵ, σ). Once we finish the K-step breadth-first search for the second component of the structure, we traverse the entire set of states reachable from the first component by procedure DFS, which is a depth-first search. Initially, we call procedure DFS(G, q) for each state $q \in \Theta$, where Θ is the set of states visited by procedure BFS. Then we consider all possible transitions and compute all possible successor states that have not been visited and make a recursive call. Note that all transitions introduced in the step are in the form of (σ, ϵ). Therefore, for each newly reached state, its distance of the second component from the initial state is the same as that of its predecessor state.

As constructed, the K-reduced TW-observer is a sub-automaton of the TW-observer. The following theorem reveals how the K-reduced TW-observer can be used to verify K-step opacity.

Theorem 7. *Let G be the system automaton, E_o be the set of observable events, and X_S be the set of secret states. Let $Obs^K_{TW}(G)$ be the K-reduced* TW-*observer of G constructed according to Algorithm* 1*. Then, G is K-step opaque w.r.t. E_o and X_S, if and only if,*

$$\forall (q_1, q_2) \in Q^K_{TW} : q_1 \cap q_2 \not\subseteq X_S \text{ or } q_1 \cap q_2 = \emptyset. \tag{22}$$

Proof. ($\Rightarrow$) By contraposition. Suppose that there exists a state $(q_1, q_2) \in Q^K_{TW}$ in $Obs^K_{TW}(G)$ such that $q_1 \cap q_2 \subseteq X_S$ and $q_1 \cap q_2 \neq \emptyset$. Since the second component q_2 is reached from the initial state within K-steps, we know that this state is reached by a string

$$t = (\epsilon, \sigma^2_1) \cdots (\epsilon, \sigma^2_k)(\sigma^1_1, \epsilon) \cdots (\sigma^1_m, \epsilon) \in \mathcal{L}(Obs^K_{TW}) \tag{23}$$

such that $|\tau_2(t)| = k \leq K$. Moreover, $Obs^K_{TW}(G)$ is a sub-automaton of $Obs_{TW}(G)$, i.e., the above string and state also exist in $Obs_{TW}(G)$. Therefore, by Theorem 6, we know that the system is not K-step opaque.

($\Leftarrow$) By contraposition. Suppose that G is not K-step opaque, which means that $\exists s_1 s_2 \in P(\mathcal{L}(G))$ such that $\hat{X}_{|s_1 s_2|-|s_2|}(s_1 s_2) \subseteq X_S$ and $|s_2| \leq K$. For the above string $s_1 s_2 \in E^*_o$, we consider the string $t' \in E^*_{TW}$ defined by Eq. (19). According to procedure BFS, we know that state $(q_{obs,0}, f_{obs,R}(X, (s_2)_R))$ is in $\Xi(d)$ for some $d \leq K$, since it can be reached by string $(\epsilon, \sigma^2_{|s_2|}) \cdots (\epsilon, \sigma^2_1)$ whose length is shorter than or equal to K and is such that all its events are of the

Algorithm 1

1: $q^K_{TW,0} \leftarrow (q_{obs,0}, X)$, $Q^K_{TW} \leftarrow \{q^K_{TW,0}\}$
2: BFS(Obs^K_{TW}, K)
3: $\Theta \leftarrow Q^K_{TW}$
4: **for all** $q \in \Theta$ **do**
5: DFS(Obs^K_{TW}, q)
6: **end for**
7: **return** $Obs^K_{TW} = (Q^K_{TW}, E_o, f^K_{TW}, q^K_{TW,0})$

8: **procedure** BFS(Obs^K_{TW}, K)
9: $D \leftarrow 0$, $\Xi(D) \leftarrow \{q^K_{TW,0}\}$
10: **for all** $0 \le D < K$ **do**
11: $\Xi(D+1) \leftarrow \emptyset$
12: **for all** $q \in \Xi(D)$ **do**
13: **for all** $\sigma \in E_o : f_{TW}(q, (\epsilon, \sigma))!$ **do**
14: **if** $f_{TW}(q, (\epsilon, \sigma)) =: q' \notin Q^K_{TW}$ **then**
15: Add transition $q \xrightarrow{(\epsilon,\sigma)} q'$ to f^K_{TW}
16: $Q^K_{TW} \leftarrow Q^K_{TW} \cup \{q'\}$
17: $\Xi(D+1) \leftarrow \Xi(D+1) \cup \{q'\}$
18: **end if**
19: **end for**
20: **end for**
21: $D \leftarrow D + 1$
22: **end for**
23: **end procedure**

24: **procedure** DFS(Obs^K_{TW}, q)
25: **for all** $\sigma \in E_o : f_{TW}(q, (\sigma, \epsilon))!$ **do**
26: **if** $f_{TW}(q, (\sigma, \epsilon)) =: q' \notin Q^K_{TW}$ **then**
27: Add transition $q \xrightarrow{(\sigma,\epsilon)} q'$ to f^K_{TW}
28: $Q^K_{TW} \leftarrow Q^K_{TW} \cup \{q'\}$
29: DFS(Obs^K_{TW}, q')
30: **end if**
31: **end for**
32: **end procedure**

form (ϵ, σ). Therefore, we know that state $(q_{obs,0}, f_{obs,R}(X, (s_2)_R))$ is in the state list Θ defined on line 3 of Algorithm 1. Moreover, state $(q_1, q_2) := (f_{obs}(q_{obs,0}, s_1), f_{obs,R}(X, (s_2)_R))$ is also reachable from $(q_{obs,0}, f_{obs,R}(X, (s_2)_R))$ in $Obs^K_{TW}(G)$, since DFS explores the entire state space reachable from the first component. Therefore, we have that

$$\emptyset \neq q_1 \cap q_2 = \hat{X}_{|s_1 s_2|-|s_2|}(s_1 s_2) \subseteq X_S$$

which completes the contrapositive proof. □

Remark 4. Similar to infinite-step opacity, when the system is not K-step opaque, based on the "⇒" direction in the proof of Theorem 7, we also can find a string $\tau_1(t)(\tau_2(t))_R \in P(\mathcal{L}(G))$, where $|\tau_2(t)| \le K$, as a cause of the violation of K-step opacity.

The next example illustrates the construction of the K-reduced TW-observer and its use it in verifying K-step opacity.

Example 6. We still consider the system G shown in Fig. 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. Let $K = 1$. The K-reduced TW-observer $Obs^K_{TW}(G)$ constructed by Algorithm 1 is shown in Fig. 3. Initially, we start procedure BFS from state (A, F), which is also the initial state of $Obs_{TW}(G)$. We have $\Xi(0) = \{(A, F)\}$. From state (A, F), we reach states (A, G) and (A, H) via strings (ϵ, a) and (ϵ, b), respectively. Then we have $\Xi(1) = \{(A, G), (A, H)\}$. Since we consider $K = 1$, we stop the breadth-first search and obtain $\Theta = \Xi(0) \cup \Xi(1)$. Then from each state in Θ, we execute a depth-first search in which only events in the form of (σ, ϵ) are considered. This yields the entire K-reduced TW-observer $Obs^K_{TW}(G)$ shown in Fig. 3. For state (B, G), which denotes state $(\{3, 4\}, \{0, 1, 2, 3, 5\})$, we have that $\{3, 4\} \cap \{0, 1, 2, 3, 5\} = \{3\} \subseteq X_S$. Therefore, by Theorem 7, we know that G is not 1-step opaque.

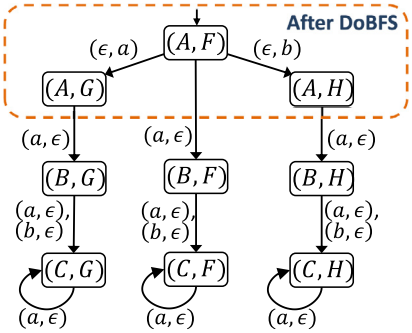


Fig. 3. The K-reduced TW-observer $Obs^K_{TW}(G)$ for the system in Fig. 1(a) for $K = 1$.

Remark 5. Let us analyze the complexity of the construction of the K-reduced TW-observer $Obs^K_{TW}(G)$. By the property of breadth-first search, we know that the number of states of the second component of the K-reduced TW-observer is bounded by $\min\{|E_o|^K, 2^{|X|}\}$. Therefore, the total complexity of this modified approach is of $O(\min\{2^{|X|}, |E_o|^K\} \times |E_o| \times 2^{|X|})$. Recall that the complexity of the algorithm in Saboori and Hadjicostis (2011b) is of $O(|E_o| \times (|E_o| + 1)^K \times 2^{|X|})$. Also, it worth noting that, in the worst, the delay K can be as large as $2^{|X|^2}$ (Saboori & Hadjicostis, 2011b). (If K is larger than $2^{|X|^2}$ then it suffices to verify infinite-step opacity.) Therefore, the TW-observer-based approach results in considerable improvement when K is relatively large.

7.3. A new upper bound for delay

In Saboori and Hadjicostis (2011b), the authors show that for any $K' > K \ge 2^{|X|^2} - 1$, K'-step opacity and K-step opacity are equivalent. In other words, it means that $2^{|X|^2} - 1$ provides the upper bound for the delay. Hereafter, we show that, in fact, this upper is conservative and it can be improved to $2^{|X|} - 2$ by using the TW-observer.

Before we state the next theorem, we introduce some necessary notions that will be used in the proof. Given the TW-observer $Obs_{TW}(G)$, we call a sequence of states and events in the form of $v = \langle q_0, \sigma_0, q_1, \ldots, q_{n-1}, \sigma_{n-1}, q_n \rangle$, $\sigma_i \in E_{TW}$, $q_i \in X_{TW}$, a *path* in $Obs_{TW}(G)$, if $q_{i+1} = f_{TW}(q_i, \sigma_i)$, $\forall i \in \{0, \ldots, n-1\}$. Now we are ready to state the main result.

Theorem 8. *For any $K' > K \ge 2^{|X|} - 2$, G is K'-step opaque, if and only if, G is K-step opaque.*

Proof. It is trivial that K'-step opacity implies K-step opacity. Hereafter, we show that K-step opacity implies K'-step opacity by contraposition. Without loss of generality, we assume that $K' = K + 1$, since it can be inductively extended to arbitrary $K' > K$.

Suppose that G is not K'-step opaque, where $K' > 2^{|X|} - 2$. This implies that there exists a string $s_1 s_2 \in P(\mathcal{L}(G))$ such that $|s_2| = K'$ and $\hat{X}_{|s_1|}(s_1 s_2) \subseteq X_S$. Let $s_1 = \sigma^1_1 \sigma^1_2 \cdots \sigma^1_{|s_1|}$ and $s_2 = \sigma^2_1 \sigma^2_2 \cdots \sigma^2_{|s_2|}$. We know that string $t = (\sigma^1_1, \epsilon) \cdots (\sigma^1_{|s_1|}, \epsilon)(\epsilon, \sigma^2_{|s_2|}) \cdots (\epsilon, \sigma^2_1)$ is in $\mathcal{L}(Obs_{TW}(G))$. This string also yields a path

$$v = \langle q_0, (\sigma^1_1, \epsilon), q_1, \ldots, q_{|s_1|-1}, (\sigma^1_{|s_1|}, \epsilon), q_{|s_1|}, (\epsilon, \sigma^2_{|s_2|}), q_{|s_1|+1}, (\epsilon, \sigma^2_{|s_2|-1}), \ldots, (\epsilon, \sigma^2_1), q_{|s_1 s_2|} \rangle$$

such that $\emptyset \neq q^1_{|s_1s_2|} \cap q^2_{|s_1s_2|} \subseteq X_S$, where $q_{|s_1s_2|} = (q^1_{|s_1s_2|}, q^2_{|s_1s_2|})$. From state $q_{|s_1|}$ to state $q_{|s_1s_2|}$, since the first component of each transition is ϵ, we know that the first components of all states from $q_{|s_1|}$ to $q_{|s_1s_2|}$ are the same. Moreover, there are only $2^{|X|} - 1$ choices for the second component of a state in $Obs_{TW}(G)$. Since the cardinality of multi-set $\{q_{|s_1|}, \ldots, q_{|s_1s_2|}\}$ is $|s_2| + 1 = K' + 1 > 2^{|X|} - 1$, we know that there exist two integers $|s_1| \leq i < j \leq |s_1s_2|$, such that $q_i = q_j$. Since states q_i and q_j are the same, we know that the existence of path v implies the existence of the following path

$$\begin{aligned} v' = \langle & q_0, (\sigma^1_1, \epsilon), q_1, \ldots, q_{|s_1|-1}, (\sigma^1_{|s_1|}, \epsilon), q_{|s_1|}, \\ & (\epsilon, \sigma^2_{|s_2|}), q_{|s_1|+1}, (\epsilon, \sigma^2_{|s_2|-1}), \ldots, \\ & (\epsilon, \sigma^2_{|s_1s_2|-i+1}), q_i, (\epsilon, \sigma^2_{|s_1s_2|-j}) \ldots, (\epsilon, \sigma^2_1), q_{|s_1s_2|} \rangle \end{aligned}$$

which means that string

$$\begin{aligned} t' = & (\sigma^1_1, \epsilon)(\sigma^1_2, \epsilon) \cdots (\sigma^1_{|s_1|}, \epsilon) \\ & (\epsilon, \sigma^2_{|s_2|}) \cdots (\epsilon, \sigma^2_{|s_1s_2|-i+1})(\epsilon, \sigma^2_{|s_1s_2|-j}) \cdots (\epsilon, \sigma^2_1) \end{aligned}$$

is in $\mathcal{L}(Obs_{TW}(G))$. Intuitively, path v' is obtained by removing the part between q_i and q_j from v. However, since

$$\tau_2(t') = |s_2| - (j - i) \leq |s_2| - 1 = K' - 1 = K.$$

Therefore, by Theorem 6, we know that G is not K-step opaque, which completes the contrapositive proof. □

The following result is an immediate consequence of Theorem 8 that improves upon the upper bound derived in Saboori and Hadjicostis (2011b).

Corollary 9. *G is infinite-step opaque if and only if it is $(2^{|X|} - 2)$-step opaque.*

Proof. The "if" part is trivial. To see the "only if" part, suppose that G is not $(2^{|X|} - 2)$-step opaque. Then, by Theorem 8, we know that G is not K'-step opaque for any K', which implies that G is not infinite-step opaque. □

8. Conclusion

We considered the two information-flow properties of infinite-step opacity and K-step opacity, that are relevant in privacy and security analysis. These properties involve smoothing, i.e., improving past state estimates on the basis of future observations. We derived a separation principle for efficiently constructing state estimates under smoothing. We then used that principle to construct a new structure, called the two-way observer, that captures in a single transition structure the information flow between past and future when analyzing the above opacity properties. New algorithms for the verification of infinite-step and K-step opacity were derived from the two-way observer. For infinite-step opacity, we showed that the proposed algorithm is more efficient and has lower complexity than the existing algorithm in the literature by lowering a factor of $2^{|X|^2}$ to a factor of $2^{|X|}$. For K-step opacity, we showed that the proposed algorithm is also more efficient and leads to significant improvement when K is relatively large. Finally, an improved upper bound for the delay in K-step opacity was also derived.

We believe that the separation principle that we established and the notion of TW-observer bring new insights into estimation problems where inferencing about the past is considered. It would be of interest to exploit the notion of TW-observer in synthesis of supervisors that enforce infinite-step or K-step opacity.

References

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., & Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dynamic Systems: Theory & Applications*, *17*(4), 425–446.

Bérard, B., Chatterjee, K., & Sznajder, N. (2015). Probabilistic opacity for Markov decision processes. *Information Processing Letters*, *115*(1), 52–59.

Bryans, J. W., Koutny, M., Mazaré, L., & Ryan, P. (2008). Opacity generalised to transition systems. *International Journal of Information Security*, *7*(6), 421–435.

Bryans, J. W., Koutny, M., & Ryan, P. (2005). Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science*, *121*, 101–115.

Cassandras, C. G., & Lafortune, S. (2008). *Introduction to discrete event systems* (second ed.). Springer.

Cassez, F. (2009). The dark side of timed opacity. In *Advances in information security and assurance* (pp. 21–30). Springer.

Cassez, F., Dubreil, J., & Marchand, H. (2012). Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design*, *40*(1), 88–115.

Chen, C.-T. (1995). *Linear system theory and design*. Oxford University Press, Inc..

Darondeau, P., Marchand, H., & Ricker, L. (2014). Enforcing opacity of regular predicates on modal transition systems. *Discrete Event Dynamic Systems: Theory & Applications*, *25*(1–2), 251–270.

Dubreil, J., Darondeau, P., & Marchand, H. (2010). Supervisory control for opacity. *IEEE Transactions on Automatic Control*, *55*(5), 1089–1100.

Falcone, Y., & Marchand, H. (2014). Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems: Theory & Applications*, 1–40.

Gruteser, M., & Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on mobile systems, applications and services* (pp. 31–42).

Jacob, R., Lesage, J.-J., & Faure, J.-M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, *41*, 135–146.

Keroglou, C., & Hadjicostis, C.N. (2013). Initial state opacity in stochastic DES. In *18th IEEE conference on emerging technologies & factory automation* (pp. 1–8).

Kobayashi, K., & Hiraishi, K. (2013). Verification of opacity and diagnosability for pushdown systems. *Journal of Applied Mathematics*, *2013*.

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, *47*(3), 496–503.

Mazaré, L. (2004). *Using unification for opacity properties. Verimag Technical Report*.

Saboori, A., & Hadjicostis, C.N. (2007). Notions of security and opacity in discrete event systems. In *46th IEEE conference on decision and control* (pp. 5056–5061).

Saboori, A., & Hadjicostis, C. N. (2011a). Coverage analysis of mobile agent trajectory via state-based opacity formulations. *Control Engineering Practice*, *19*(9), 967–977.

Saboori, A., & Hadjicostis, C. N. (2011b). Verification of K-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, *8*(3), 549–559.

Saboori, A., & Hadjicostis, C. N. (2012a). Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Transactions on Automatic Control*, *57*(5), 1155–1165.

Saboori, A., & Hadjicostis, C. N. (2012b). Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, *57*(5), 1265–1269.

Saboori, A., & Hadjicostis, C. N. (2013). Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, *246*, 115–132.

Saboori, A., & Hadjicostis, C. N. (2014). Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control*, *59*(1), 120–133.

Takai, S., & Kumar, R. (2009). Verification and synthesis for secrecy in discrete-event systems. In *American control conference* (pp. 4741–4746).

Takai, S., & Oka, Y. (2008). A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE Journal of Control, Measurement, and System Integration*, *1*(4), 307–311.

Tong, Y., Li, Z., Seatzu, C., & Giua, A. (2016). Verification of state-based opacity using Petri nets. *IEEE Transactions on Automatic Control*, http://dx.doi.org/10.1109/TAC.2016.2620429.

Varaiya, P., & Kumar, P. R. (1986). *Stochastic systems: Estimation, identification and adaptive control*. Englewood Cliffs, NJ: Prentice-Hall.

Wu, Y.-C., & Lafortune, S. (2013). Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems: Theory & Applications*, *23*(3), 307–339.

Wu, Y.-C., & Lafortune, S. (2014). Synthesis of insertion functions for enforcement of opacity security properties. *Automatica*, *50*(5), 1336–1348.

Wu, Y.-C., Raman, V., Lafortune, S., & Seshia, S.A. (2016). Obfuscator synthesis for privacy and utility. In *Proceedings of the 8th NASA formal methods symposium* (pp. 133–149).

Wu, Y.-C., Sankararaman, K.A., & Lafortune, S. (2014). Ensuring privacy in location-based services: An approach based on opacity enforcement. In *12th int. workshop on discrete event systems*, vol. 12 (pp. 33–38).

Yin, X., & Lafortune, S. (2015a). A general approach for solving dynamic sensor activation problems for a class of properties. In *54th IEEE conference on decision and control* (pp. 3610–3615).

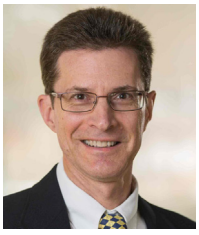
Yin, X., & Lafortune, S. (2015b). A new approach for synthesizing opacity-enforcing supervisors for partially-observed discrete-event systems. In *American control conference* (pp. 377–383).

Yin, X., & Lafortune, S. (2016a). On two-way observer and its application to the verification of infinite-step and K-step opacity. In *13th int. workshop on discrete event systems* (pp. 361–366).

Yin, X., & Lafortune, S. (2016b). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, *61*(8), 2140–2154.

Zhang, B., Shu, S., & Lin, F. (2015). Maximum information release while ensuring opacity in discrete event systems. *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1067–1079.

Xiang Yin was born in Anhui, China, in 1991. He received the B.Eng degree from Zhejiang University in 2012; the M.S. degree from the University of Michigan, Ann Arbor, in 2013; and the Ph.D degree from the University of Michigan, Ann Arbor, in 2017, all in electrical engineering. His research interests include supervisory control of discrete-event systems, model-based fault diagnosis, formal methods, security and their applications to cyber and cyber–physical systems. He received the Outstanding Reviewer Award from Automatica in 2016, the Outstanding Reviewer Award from IEEE Transactions on Automatic Control in 2017 and the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016. He is the co-chair of the IEEE CSS Technical Committee on Discrete Event Systems.



Stéphane Lafortune received the B.Eng degree from Ecole Polytechnique de Montréal in 1980; the M.Eng degree from McGill University in 1982; and the Ph.D degree from the University of California at Berkeley in 1986, all in electrical engineering. Since September 1986, he has been with the University of Michigan, Ann Arbor, where he is a professor of electrical engineering and computer science.

He is a Fellow of the IEEE (1999). He received the Presidential Young Investigator Award from the National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the Control Systems Society of the IEEE in 1994 (for a paper co-authored with S.-L. Chung and F. Lin) and in 2001 (for a paper co-authored with G. Barrett). His research interests are in discrete event systems and include multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer and software systems. He is the lead developer of the software package UMDES and co-developer of DESUMA with L. Ricker. He co-authored, with C. Cassandras, the textbook Introduction to Discrete Event Systems — Second Edition (Springer, 2008). He is an editor-in-chief of the Journal of Discrete Event Dynamic Systems: Theory and Applications.