2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

To Transmit or Not to Transmit: Optimal Sensor Scheduling for Remote State Estimation of Discrete-Event Systems

Yingying Liu, Xiang Yin and Shaoyuan Li

***Abstract*— This paper considers the problem of optimal sensor scheduling for remote state estimation of discrete-event systems. In this setting, the sensors observe events from the plant and transmit them to the receiver or estimator selectively. The transmission mechanism decides to transmit the observed information or not according to an *information transmission policy*. The receiver needs to have sufficient information for the purpose of decision-making. To solve this problem, we first construct a non-deterministic dynamic observer that contains all feasible information transmission policies. Then we show that the information updating rule of the dynamic observer indeed yields the state estimate from the receiver's point of view. Finally, we propose an approach to extract a specific information transmission policy, realized by a finite-state automaton, from the dynamic observer while satisfying some desired observation properties. A running example is provided to illustrate the proposed procedures.**

I. INTRODUCTION

Discrete-event systems (DES) is an importance class of engineering systems such as manufacturing systems and autonomous robots [1], [2], [3]. In real world systems, events are sometimes unobservable due to the limited observation capacity. Therefore, the estimation of states for the system is an important problem. Most works in partially-observed DES considers static observations, where all sensor readings are assumed to be sent to the user/receiver. However, in many modern applications such as remote control [4], or remote estimation [5], the sensors and the user of the information are physically different and located remotely. Therefore, the sensors can choose to transmit or not transmit their observations [6], [7].

In this paper, we consider a remote state estimation problem of discrete-event systems. As shown in Fig. 1, we consider the scenario where the system makes observations online through its sensors. The sensors can be turned ON/OFF dynamically by a transmission switch, where the switch decides whether to transmit its observation to a receiver or not, according to *an information transmission policy*. It is precisely because of such property that we do not need to transmit one observable event all the time and only need to transmit it when necessary (to satisfy some observational property).

Then the receiver makes control decisions for the system based on the transmitted information. Our objective is to

This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012) and the National Key Research and Development Program of China (2018AAA0101700). The authors are with the Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {yingyingliu611,yinxiang,syli}@sjtu.edu.cn.

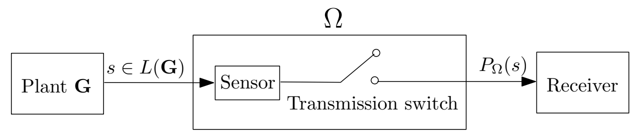

Fig. 1. Architecture of transmission mechanism, where Ω denotes the transmission transmission policy and P_Ω denotes the information mapping under policy Ω (see precise definitions in Section II).

synthesize an information transmission policy such that some given property holds. Instead of investigating the enforcement of specific objectives, e.g., control [8] or diagnosis [9], in this paper, we consider a particular class of properties called Information-State-based (IS-based) properties [10]. Roughly speaking, an IS-based property is a property that only depends on the current local information of the system.

To construct such a transmission mechanism, we synthesize a non-deterministic dynamic observer that enforces state disambiguation and contains all feasible information transmission policies. It is ensured that the synthesis problem of a feasible information transmission policy is always solvable. We prove that the information updating rule of the dynamic observer indeed yields the state estimate of the receiver. We also propose a method to extract a specific information transmission policy from the dynamic observer while ensuring the given IS-based properties. Moreover, to reduce sensor-related costs caused by energy, bandwidth, or security constraints, we require that the sensors transmit as few events as possible.

Our work is closely related to the so-called dynamic sensor activation proposed in [11], [12], [13], [14], [15], [16]. In this context, the sensors and the information user can the same. By contrast, this paper considers the remote estimation problem where the sensor and the receiver are located at a remote distance and thus be more complex and challenging. Also, authors in [17], [18] investigate how to release the maximum information to the public by a controller while ensuring opacity. This setting is similar to us. However, here we consider the IS-based property, which is not restricted to opacity and can formulate utilities such as diagnosability, detectability, and distinguishability.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. System Model

We consider a DES modeled by a deterministic finite-state automaton (DFA) $\mathbf{G} = (Q, \Sigma, \delta, q_0)$, where Q is a finite set of states, Σ is a finite set of events, $\delta\colon Q \times \Sigma \to Q$ is a (partial) transition function, $q_0 \in Q$ is the initial state. In the usual way, δ can be extended to $\delta\colon Q \times \Sigma^* \to Q$, where Σ^* is the set of all finite-length *strings*, including the

978-1-6654-5196-3/$31.00 ©2022 AACC

empty string ε. The *generated behavior* of $\mathbf{G}$ is language $L(\mathbf{G}) = \{s \in \Sigma^* : \delta(q_0, s)!\}$, where $\delta(q_0, s)!$ means that $\delta(q_0, s)$ is defined. We denote $\Sigma_q = \{\sigma \in \Sigma : \delta(q, \sigma)!\}$ the set of events that are defined at state q.

A string $s_1 \in \Sigma^*$ is a *prefix* of $s \in \Sigma^*$, written as $s_1 \leq s$, if there is a string $s_2 \in \Sigma^*$ such that $s_1 s_2 = s$. The length of a string s is denoted by $|s|$. The *prefix closure* of a language L is the set $\overline{L} = \{s \in \Sigma^* : \exists t \in L \text{ s.t. } s \leq t\}$. For a natural number n, let $[1, n] = \{1, \ldots, n\}$ denote the set of all natural numbers from 1 to n. Let $\Sigma_o \subseteq \Sigma$, we denote by $P : \Sigma_o^* \to \Sigma^*$ the standard natural projection from Σ_o to Σ.

B. Problem Formulation

We consider the scenario where the system $\mathbf{G}$ is equipped with sensors that monitor the global system. As shown in Fig. 1, the sensors observe global events. They can be turned ON/OFF dynamically by a transmission switch, where the transmission switch decides, based on the observation history, whether to transmit this observation to a receiver or not. Here we consider a generic receiver, and it can be, e.g., a supervisor or a diagnoser, depending on the specific application. The receiver makes control decisions for the system based on the obtained information. Such a decision mechanism is formalized as an *information transmission policy*. Let $\Sigma_o \subseteq \Sigma$ be the set of events observed by sensors. In this paper, we suppose all events are observable by sensors (full observation), i.e., $\Sigma_o = \Sigma$. The information transmission policy is defined by

$$\Omega : \Sigma^* \Sigma \to \{Y, N\},$$

where Y and N are *information transmission labels*. Specifically, for each observation $s\sigma \in \Sigma^*\Sigma$, Ω will decide whether to transmit the observation of σ, i.e., $\Omega(s\sigma) = Y$, or not $\Omega(s\sigma) = N$. The above definition of the information transmission policy is history-dependent. Note that the observability of an event with different history may be different. In practice, the information transmission policy needs to be implemented in finite memory, which can be represented as a pair (a finite transducer)

$$\Omega = (\mathbf{A}, \mathcal{L}), \tag{1}$$

where $\mathbf{A} = (X, \Sigma_A, \eta, x_0)$ is a DFA, called a *sensor automaton*, such that (i) $L(\mathbf{A}) = \Sigma_A^*$; and (ii) $\forall x \in X, \sigma \notin \Sigma_o : \eta(x, \sigma) = x$, and $\mathcal{L} : X \times \Sigma_A \to \{Y, N\}$ is a labelling function that determines whether the current observable event is transmitted or not. Here, we assume the event domain of $\mathbf{A}$ is Σ for the sake of simplicity, but it can only update its sensor state upon the occurrences of its observable events in Σ_A. Also, for any $\sigma \in \Sigma_A$, $\mathcal{L}(x, \sigma) = Y$ means that the occurrence of event σ will be transmitted at state x, while $\mathcal{L}(x, \sigma) = N$ represents the opposite. Hereafter in the paper, an information transmission policy will be considered as a pair $\Omega = (\mathbf{A}, \mathcal{L})$ rather than a language-based mapping.

The projection based on a given information transmission policy Ω is recursively defined by $P_\Omega : L(\mathbf{G}) \to \mathbf{\Sigma}^*$

$$P_\Omega(\epsilon) = \epsilon; P_\Omega(s\sigma) = \begin{cases} P_\Omega(s)\sigma, & \text{if } \Omega(s\sigma) = Y; \\ P_\Omega(s), & \text{if } \Omega(s\sigma) = N. \end{cases}$$

For any string $s \in L(\mathbf{G})$ generated by the system, we define $\mathcal{E}_\Omega^G(s) := \{\delta(q_0, t) \in Q : \exists t \in L(\mathbf{G}) \text{ s.t. } P_\Omega(s) = P_\Omega(t)\}$ as the *state estimate* of the receiver. Clearly, for strings $s, t \in L(\mathbf{G})$, if $P_\Omega(s) = P_\Omega(t)$, then $\mathcal{E}_\Omega^G(s) = \mathcal{E}_\Omega^G(t)$. We define $P_\Omega^{-1}(s) = \{s' \in \Sigma^* : P_\Omega(s) = P_\Omega(s')\}$.

In this work, instead of considering specific objectives, e.g., control or diagnosis, we consider a particular class of properties called Information-State-based (IS-based) properties [10]. The notion of an information state is defined as a subset $IS \subseteq Q$ of states of $\mathbf{G}$ and the set of all information states is denoted by $I = 2^Q$.

Definition 1: Given an automaton $\mathbf{G}$, an IS-based property φ w.r.t. $\mathbf{G}$ is a function $\varphi : 2^Q \to \{0, 1\}$, where $\forall i \in 2^Q$, $\varphi(i) = 1$ means that i satisfies this property. We say that sublanguage $L \subseteq L(\mathbf{G})$ satisfies φ w.r.t. $\mathbf{G}$, which is denoted by $L \models_{\mathbf{G}} \varphi$, if $\forall s \in L : \varphi(R_G(s, L)) = 1$, where $R_G(s, L) = \{\delta(q_0, t) \in Q : \exists t \in L(\mathbf{G}) \text{ s.t. } P(s) = P(t)\}$.

For more details about IS-based property, the reader is referred to [10]. We will employ the distinguishability property as a running example to illustrate the results.

Our objective is to synthesize an information transmission policy such that some given property holds. We define the Information Transmission Problem (IT Problem) as follows.

Porblem 1: Given a plant $\mathbf{G} = (Q, \Sigma, \delta, q_0)$ and a IS-based property $\varphi : 2^Q \to \{0, 1\}$. Find an information transmission policy $\Omega = (\mathbf{A}, \mathcal{L})$ s.t.

(i) $L(\mathbf{A})$ satisfies φ w.r.t. $\mathbf{G}$, i.e., $L(\mathbf{A}) \models_{\mathbf{G}} \varphi$;

(ii) the sensors transmit as few events as possible.

III. A General Most Comprehensive Dynamic Observer

For implementation purposes, we firstly attach the information transmission labels $\{Y, N\}$ to the system $\mathbf{G}$. Then we construct a dynamic observer that contains all possible information transmission policies. Finally, we formally show that the information updating rule of the dynamic observer indeed yields the state estimate of the receiver.

A. Attach information transmission labels to the system

We attach the information transmission labels to the system $\mathbf{G}$, called *labelled system*, which can be represented as a nondeterministic finite-state automaton (NFA)

$$\mathbf{G}_{ag} = (\tilde{Q}, \Sigma, f, \tilde{Q}_0), \tag{2}$$

where

- $\tilde{Q} = Q \times \{Y, N\}^{|\Sigma|}$ is the set of states;
- Σ is the set of events;
- $f : \tilde{Q} \times \Sigma \to 2^{\tilde{Q}}$ is the partial transition function defined by: for any $\tilde{q}, \tilde{q}' \in \tilde{Q}$ and $\sigma \in \Sigma$, $\tilde{q}' \in f(\tilde{q}, \sigma)$ if $q' = \delta(q, \sigma)$, i.e., the labels of $\tilde{q}$ is free to change.
- $\tilde{Q}_0 = \{q_0\} \times \{Y, N\}^{|\Sigma|}$.

As shown above, the state space of $\mathbf{G}_{ag}$ is defined by $\tilde{Q} = Q \times \{Y, N\}^{|\Sigma|}$. For each state $\tilde{q} \in \tilde{Q}$, we list all possible transmission decisions/labels of all events in Σ for $\tilde{q}$. Specifically, for each event σ, we have two transmission

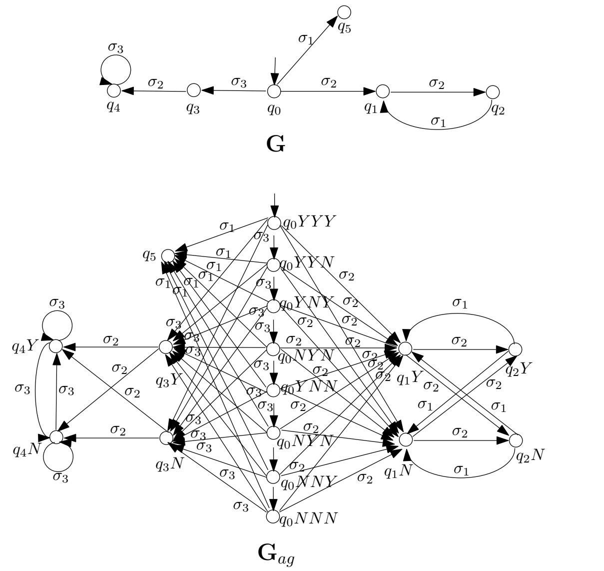

Fig. 2. System $\mathbf{G}$ and labelled system $\mathbf{G}_{ag}$. For simplification, instead of listing the labels of all events to each state, we list the labels of events that are defined at each state; namely, only $2^{|\Sigma_q|}$ labels are listed at state q.

decisions: transmit Y and not transmit N. These two decisions are attached at $\tilde{q}$. Similarly, We list the transmission decisions of each event at each state in $\mathbf{G}_{ag}$.

Example 1: Given plant $\mathbf{G}$ in Fig. 2, we attach the information transmission labels $\{Y, N\}$ to $\mathbf{G}$ to construct $\mathbf{G}_{ag}$ as defined by (2). The initial states of $\mathbf{G}_{ag}$ are $\tilde{q}_0 = \{q_0\} \times \{Y, N\}^{|\Sigma_{q_0}|} = \{q_0\} \times \{Y, N\}^3 = \{q_0YYY, q_0YYN, q_0YNY, q_0NYY, q_0YNN, q_0NNY, q_0NNY, q_0NNN\}$ since there have three events (σ_1, σ_2, and σ_3) to be defined at q_0. Similarly, we get the state space $\tilde{Q} = \{q_0YYY, q_0YYN, q_0YNY, q_0NYY, q_0YNN, q_0NNY, q_0NNY, q_0NNN, q_1Y, q_1N, q_2Y, q_2N, q_3Y, q_3N, q_4Y, q_4N, q_5\}$. The constructed $\mathbf{G}_{ag}$ is shown in Fig. 2.

B. Construction of the dynamic observer

To give all possible information policies, we construct a dynamic observer for $\mathbf{G}_{ag}$, which needs the following definitions. In order to clearly give the transmission policy of each defined event, we employ $\mathbb{L}$ to denote the set of all functions $l : \Sigma \to \{Y, N\}$. The function l maps events in Σ to the labels $\{Y, N\}$. Specifically, $l(\sigma) = Y$ means the event σ is transmitted by the sensor, and $l(\sigma) = N$ represents the opposite. For any two states $\tilde{q} = (q, l), \tilde{q}' = (q', l') \in \mathbf{G}_{ag}$ with $l, l' \in \mathbb{L}$, an *unobservable path* from $\tilde{q}$ to $\tilde{q}'$ is a sequence in $\mathbf{G}_{ag}$

$$\mathbf{p} = (q_1, l_1) \xrightarrow{\sigma_1} (q_2, l_2) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (q_{n+1}, l_{n+1}), \qquad (3)$$

where $\tilde{q} = (q_1, l_1)$ and $\tilde{q}' = (q_{n+1}, l_{n+1})$ such that, for all $i = 1, \ldots, n : l_i = l(\sigma_i) = N$. We denote $\mathsf{u\text{-}path}(\tilde{q}, \tilde{q}')$ the set of unobservable paths (u-path) from $\tilde{q}$ to $\tilde{q}'$ and say $\tilde{q}'$ is *reached unobservably* from $\tilde{q}$ by $\mathsf{u\text{-}path}(\tilde{q}, \tilde{q}')$. Thus the set of unobservable paths from $\tilde{q}$ is defined by

$$\mathsf{Up}(\tilde{q}) = \bigcup_{\tilde{q}' \in \tilde{Q}} \mathsf{u\text{-}path}(\tilde{q}, \tilde{q}').$$

For any $\tilde{q}' \in \tilde{Q}$ reached unobservably from $\tilde{q}$ by $\mathbf{p}$, we denote

$$\hat{Q}(\mathbf{p}) = \{(q_1, l_1), (q_2, l_2), \cdots, (q_{n+1}, l_{n+1})\}$$

the set of states reached unobservably from $\tilde{q}$ to $\tilde{q}'$ by $\mathbf{p}$. The set of states that are unobservable reached from $\tilde{q}$ thus can be obtained by

$$S(\tilde{q}) = \bigcup_{\mathbf{p} \in \mathsf{Up}(\tilde{q})} \hat{Q}(\mathbf{p}) \qquad (4)$$

A set $\imath$ is said to be *unobservable reach-closed* if for any states $(q_i, l_i) \in \imath$, we have

$$l(q_i, \sigma_i) = N : \ \exists (q_j, l_j) \in \imath \ s.t., f(\tilde{q}_i, \sigma_i) = \tilde{q}_j.$$

That is to say, for any state (q_i, l_i) in $\imath$, if the event σ_i defined at (q_i, l_i) will not be transmitted, i.e., $l(q_i, \sigma_i) = N$, then there must exists a state (q_j, l_j) in $\imath$ such that (q_j, l_j) is reachable by σ_i from (q_i, l_i).

Consider the collection of all subsets of $S(\tilde{q})$ that are unobservable reach-closed:

$$\mathcal{UC}(\tilde{q}) = \{\imath \subseteq S(\tilde{q}) : \imath \text{ is unobservable reach-closed}\}. \quad (5)$$

It is straightforward to verify that $\mathcal{UC}(\tilde{q})$ is nonempty ($\tilde{q}$ belongs) and is closed under arbitrary unions.

Let $\imath = \{\tilde{q}_1, \ldots, \tilde{q}_n\}$. The subsets that are unobservable reach-closed are defined by

$$\mathcal{UC}(\imath) = \bigcup_{\tilde{q} \in \imath} \mathcal{UC}(\tilde{q}).$$

Moreover, there may be a decision conflict because for each $\sigma \in \Sigma$ we can only choose either Y or N for each $\sigma \in \Sigma$. Therefore, some states cannot occur simultaneously in $S(\tilde{q})$. We say two sequences

$$\mathbf{p} = (q_1, l_1) \xrightarrow{\sigma_1} (q_2, l_2) \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_n} (q_{n+1}, l_{n+1})$$
$$\mathbf{p}' = (q_1', l_1') \xrightarrow{\sigma_1'} (q_2', l_2') \xrightarrow{\sigma_2'} \cdots \xrightarrow{\sigma_m'} (q_{m+1}', l_{m+1}')$$

are non-conflicting if

$$\forall i \leq \min\{n, m\} : (q_i, l_i) \neq (q_i', l_i') \Rightarrow \sigma_1 \ldots \sigma_i \neq \sigma_1' \ldots \sigma_i'$$

We say $\imath \subseteq S(\tilde{q})$ is non-conflicting if there exists a collection of u-paths $\mathbf{P} \subseteq \mathsf{Up}(\tilde{q})$ such that each pair of upaths in it are non-conflicting and $\imath = \bigcup_{\mathbf{p} \in \mathbf{P}} \hat{Q}(\mathbf{p})$.

Again consider the collection of all subsets of $S(\tilde{q})$ that are non-conflicting:

$$\mathcal{NC}(\tilde{q}) = \{\imath \subseteq S(\tilde{q}) : \imath \text{ is non-conflicting}\}.$$

It can be verified that $\mathcal{NC}$ is nonempty and is closed under arbitrary intersections. Similarly, let $\imath = \{\tilde{q}_1, \ldots, \tilde{q}_n\}$. The subsets of $\imath$ that are non-conflicting are defined by

$$\mathcal{NC}(\imath) = \bigcup_{\tilde{q} \in \imath} \mathcal{NC}(\tilde{q}).$$

Let

$$maxS(\imath) = \mathcal{UC}(\imath) \cap \mathcal{NC}(\imath) \qquad (6)$$

be the maximal subset of $S(\imath)$ that is unobservable reach-closed and non-conflicting. Note that $maxS(\imath)$ is not a singleton because we may have different maximal choices that are conflicting.

Let $\tilde{q} = (q, l)$ be an augmented state and $\sigma \in \Sigma$ be an observable event. $NX_\sigma(\tilde{q})$ is defined if $l(\sigma) = Y$ and $f(q, \sigma)!$. Furthermore, we have

$$NX_\sigma(\tilde{q}) = \{\tilde{q}' \in \tilde{Q} : \tilde{q}' \in f(\tilde{q}, \sigma)\} \quad (7)$$

For $\imath = \{\tilde{q}_1, \ldots, \tilde{q}_n\}$, we define

$$NX_\sigma(\imath) = \{\{\tilde{q}_1', \ldots, \tilde{q}_n'\} \in 2^{\tilde{Q}} : \forall i \in [1, n],\ \tilde{q}_i' \in f(\tilde{q}_i, \sigma)\}. \quad (8)$$

We define the set of states that can be reached unobservably from some state in $\imath$ by $UR(\imath) = maxS(\imath)$.

Now we are ready to construct the dynamic observer of $\mathbf{G}_{ag}$, which is defined as a new NFA

$$Obs(\mathbf{G}_{ag}) = (Z, \Sigma, \xi, Z_0), \quad (9)$$

where

- $Z \subseteq 2^{\tilde{Q}}$ is the set of states;
- Σ is the set of events;
- $\xi : Z \times \Sigma \to 2^Z$ is the partial transition function defined by: for any $z \in Z, \sigma \in \Sigma$, we have
$$\xi(z, \sigma) = UR(NX_\sigma(z)) = \bigcup_{z' \in NX_\sigma(z)} maxS(z'); \quad (10)$$
- $Z_0 = UR(\tilde{Q}_0)$ is the set of initial states.

The initial states Z_0 is defined by $UR(\tilde{Q}_0) = \bigcup_{\tilde{q}_0 \in \tilde{Q}_0} maxS(\tilde{q}_0)$. $UR(z)$ is the set of states that can be reached unobservably from some state in z and is unobservable reach-closed and non-conflicting. Since $UR(z) = maxS(z)$ and $maxS(z)$ may not a singleton, $UR(z)$ may not singleton as well. $NX_\sigma(z)$ is the set of states that can be reached from some state $\tilde{q}$ in z immediately by event σ, i.e., $f(\tilde{q}, \sigma)$. Similarly, $f(\tilde{q}, \sigma)$ may not singleton because the future states of $\tilde{q}$ reached by σ are attached with different labels.

As defined above, we employ $NX_\sigma(z)$ to update the states when the event σ is transmitted at state z. After obtaining the updated state, we use UR to compute the states that are reached unobservably from states in $NX_\sigma(z)$. Since $UR(z) = maxS(z) = \mathcal{UC}(z) \cap \mathcal{NC}(z)$, we need to find all the unobservable paths $\mathbf{p}$ as given in (3) firstly. Then by deleting the paths whose generated states are not unobservable reach-closed and conflicting, we get the paths that satisfy (6). It is ensured that all states in $Obs(\mathbf{G}_{ag})$ are unobservable reach-closed and non-conflicting. In addition, $Obs(\mathbf{G}_{ag})$ contains all possible information transmission policies. An illustrative example is given in the following.

Example 2: Let us consider the system $\mathbf{G}$ and the labelled system $\mathbf{G}_{ag}$ in Fig. 2. We employ this example to illustrate the procedure of synthesizing the state set of $Obs(\mathbf{G}_{ag})$. In this example, we only show partial paths started from the initial state q_0NNY. The cases started from other initial states are similar.

Initially, we need to compute the initial states started from q_0NNY. We have $UR(q_0NNY) = maxS(q_0NNY) = \mathcal{UC}(q_0NNY) \cap \mathcal{NC}(q_0NNY)$. By (4) we have $S(q_0NNY) = \{\hat{Q}(\mathbf{p}_1), \hat{Q}(\mathbf{p}_2), \ldots, \hat{Q}(\mathbf{p}_5)\}$, where $\mathbf{p}_1 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1Y)$, $\mathbf{p}_2 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1N)$, $\mathbf{p}_3 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1N) \xrightarrow{\sigma_2} (q_2Y)$, $\mathbf{p}_4 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1N) \xrightarrow{\sigma_2} (q_2N) \underset{\sigma_2}{\overset{\sigma_1}{\rightleftharpoons}} (q_1N)$, $\mathbf{p}_5 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1N) \xrightarrow{\sigma_2} (q_2N) \underset{\sigma_2}{\overset{\sigma_1}{\rightleftharpoons}} (q_1N) \xrightarrow{\sigma_2} (q_2Y)$, and $\mathbf{p}_6 = (q_0NNY) \xrightarrow{\sigma_1,\sigma_2} (q_5, q_1N) \xrightarrow{\sigma_2} (q_2N) \xrightarrow{\sigma_1} (q_1N) \underset{\sigma_1}{\overset{\sigma_2}{\rightleftharpoons}} (q_2N) \xrightarrow{\sigma_2} (q_1Y)$. $\hat{Q}(\mathbf{p}_2)$ is not unobservable reach-closed since for state $q_1N \in \hat{Q}(\mathbf{p}_2)$, there does not exist a state in $\hat{Q}(\mathbf{p}_2)$ s.t. q_1N can reach it. We thus get that $\mathcal{UC}(q_0NNY) = \{\hat{Q}(\mathbf{p}_1), \hat{Q}(\mathbf{p}_3), \hat{Q}(\mathbf{p}_4), \hat{Q}(\mathbf{p}_5), \hat{Q}(\mathbf{p}_6)\}$. It can be verified that all the paths in $\mathcal{UC}(q_0NNY)$ are non-conflicting, we thus get that $UR(q_0NNY) = maxS(q_0NNY) = \{\hat{Q}(\mathbf{p}_1), \hat{Q}(\mathbf{p}_3), \hat{Q}(\mathbf{p}_4), \hat{Q}(\mathbf{p}_5), \hat{Q}(\mathbf{p}_6)\} = \{\{q_0NNY, q_5, q_1Y\}, \{q_0NNY, q_5, q_1N, q_2Y\}, \{q_0NNY, q_5, q_1N, q_2N\}, \{q_0NNY, q_5, q_1N, q_2N, q_1Y\}, \{q_0NNY, q_5, q_1N, q_2N, q_2Y\}\}$.

Then we employ the initial state $z_0 = \{q_0NNY, q_5, q_1Y\}$ in $maxS(q_0NNY)$ to illustrate how to update the states in $Obs(\mathbf{G}_{ag})$. Since there have two events σ_2, σ_3 that are defined at $\{q_0NNY, q_5, q_1Y\}$, by (10) we have transitions: (i) $\xi(z_0, \sigma_2) = UR(NX_{\sigma_2}(z_0)) = \bigcup_{z' \in NX_{\sigma_2}(z_0)} maxS(z')$. By (8) we have $NX_{\sigma_2}(z_0) = \{\{\tilde{q}_2'\} : \tilde{q}_2' \in f(q_0NNY, \sigma_2)\} = \{\{q_2Y\}, \{q_2N\}\}$. Then $\xi(z_0, \sigma_2) = maxS(\{q_2Y\}) \cup maxS(\{q_2N\})$. Since $\{q_2Y\}$ is labeled with Y, it is directly obtained that $maxS(\{q_2Y\}) = \{q_2Y\}$. For $\{q_2N\}$ we have $maxS(\{q_2N\}) = \{\hat{Q}(\mathbf{p}_1'), \hat{Q}(\mathbf{p}_2'), \hat{Q}(\mathbf{p}_3'), \hat{Q}(\mathbf{p}_4')\}$, where $\mathbf{p}_1' = (q_2N) \xrightarrow{\sigma_1} (q_1Y)$, $\mathbf{p}_2' = (q_2N) \underset{\sigma_2}{\overset{\sigma_1}{\rightleftharpoons}} (q_1N)$, $\mathbf{p}_3' = (q_2N) \underset{\sigma_2}{\overset{\sigma_1}{\rightleftharpoons}} (q_1N) \xrightarrow{\sigma_2} (q_2Y)$, and $\mathbf{p}_4' = (q_2N) \xrightarrow{\sigma_1} (q_1N) \underset{\sigma_2}{\overset{\sigma_1}{\rightleftharpoons}} (q_2N) \xrightarrow{\sigma_2} (q_1Y)$, We thus obtain $\xi(z_0, \sigma_2) = maxS(\{q_2Y\}) \cup maxS(\{q_2N\}) = \{\{q_2Y\}, \{q_2N, q_1Y\}, \{q_1N, q_2N\}, \{q_1N, q_2N, q_1Y\}, \{q_1N, q_2N, q_2Y\}\}$. (ii) $\xi(z_0, \sigma_3) = UR(NX_{\sigma_3}(z_0)) = \bigcup_{z' \in NX_{\sigma_3}(z_0)} maxS(z')$. By (8) we have $NX_{\sigma_3}(z_0) = \{\{\tilde{q}_3'\} : \tilde{q}_4' \in f(q_0NNY, \sigma_3)\} = \{\{q_3Y\}, \{q_3N\}\}$. Similarly, we have $maxS(\{q_3N\}) = \{\hat{Q}(\mathbf{p}_1''), \hat{Q}(\mathbf{p}_2''), \hat{Q}(\mathbf{p}_3'')\}$, where $\mathbf{p}_1'' = (q_3N) \xrightarrow{\sigma_3} (q_4Y)$, $\mathbf{p}_2'' = (q_3N) \xrightarrow{\sigma_2} (q_4N) \underset{\sigma_3}{\overset{\sigma_3}{\rightleftharpoons}} (q_4N)$, and $\mathbf{p}_3'' = (q_3N) \xrightarrow{\sigma_2} (q_4N) \underset{\sigma_3}{\overset{\sigma_3}{\rightleftharpoons}} (q_4N) \xrightarrow{\sigma_3} (q_4Y)$. We thus get that $\xi(z_0, \sigma_3) = maxS(\{q_3Y\}) \cup maxS(\{q_3N\}) = \{\{q_3Y\}, \{q_3N, q_4Y\}, \{q_3N, q_4N\}, \{q_3N, q_4N, q_4Y\}\}$.

The other states can be computed in the similar way. The obtained dynamic observer $Obs(\mathbf{G}_{ag})$ is shown in Fig. 3.

C. State Estimate

As mentioned in Section III-A, all possible information transmission policies are listed in $\mathbf{G}_{ag}$. These information transmission policies can be restricted to a specific policy $\Omega = (\mathbf{A}, \mathcal{L})$ by producting them. The product of $\mathbf{A} =$

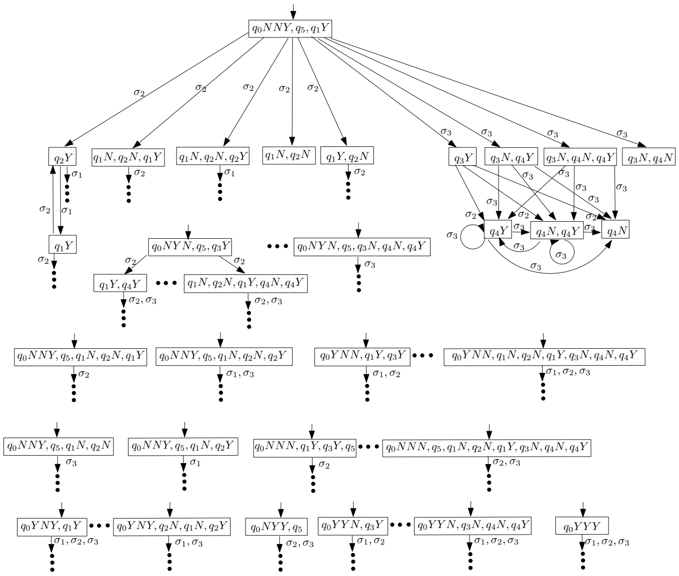

Fig. 3. The dynamic observer $Obs(\mathbf{G}_{ag})$

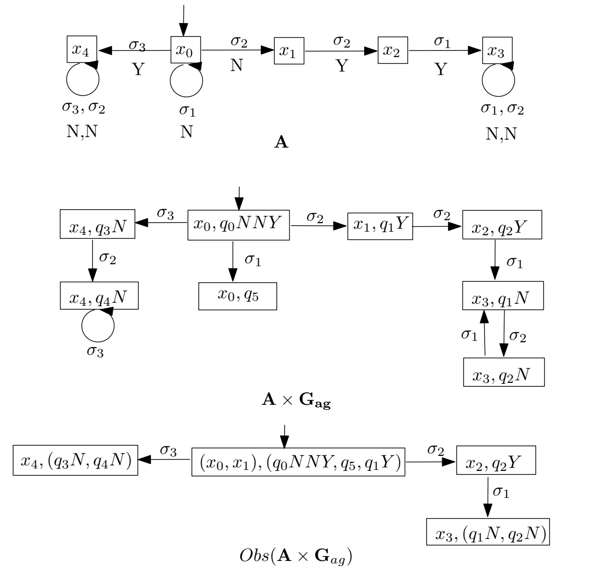

Fig. 4. Sensor automata $\mathbf{A}$, the product automata $\mathbf{A} \times \mathbf{G}_{ag}$ and the observer $Obs(\mathbf{A} \times \mathbf{G}_{ag})$.

(X, Σ, η, x_0) and $\mathbf{G}_{ag} = (\tilde{Q}, \Sigma, f, \tilde{Q}_0)$ is defined by $\mathbf{A} \times \mathbf{G}_{ag} = (Y, \Sigma, \theta, y_0)$, where

- $Y \subseteq X \times \tilde{Q}$ is the set of states,
- Σ is the set of events,
- $\theta : Y \times \Sigma \to Y$ is the partial transition function defined by: for any $y = (x, \tilde{q}), y' = (x', \tilde{q}') \in Y, \sigma \in \Sigma$, $\forall \sigma' \in \Sigma_{\tilde{q}'}$, we have $y' = \theta(y, \sigma)$ iff

$$x' = \eta(x, \sigma) \wedge \tilde{q}' \in f(\tilde{q}, \sigma) \wedge l(\sigma') = \mathcal{L}(x', \sigma') \quad (11)$$

- $y_0 = (x_0, \tilde{q}_0)$ with $l(\sigma) = \mathcal{L}(x_0, \sigma)$, for all $\sigma \in \Sigma_{\tilde{q}_0}$, is the initial states.

To estimate the states of the system under the given information transmission policy Ω, we construct the observer of $\mathbf{A} \times \mathbf{G}_{ag}$, which is defined by $Obs(\mathbf{A} \times \mathbf{G}_{ag}) = (Y', \Sigma, \theta', y_0')$, where

- $Y' \subseteq 2^Y = 2^{X \times \tilde{Q}}$ is the set of states,
- Σ is the set of events,
- $\theta' : Y' \times \Sigma \to Y'$ is the partial transition function defined by: for any $\imath, \imath' \in Y', \sigma \in \Sigma$, we have $\imath' = \theta'(\imath, \sigma)$ iff

$$\imath' = UR'(NX_\sigma'(\imath)) \quad (12)$$

where for any $\imath \in 2^Y$ we have

$$NX_\sigma'(\imath) = \{y' \in Y : \exists\, y \in \imath \ s.t.\ \theta(y, \sigma) = y'\}$$
$$UR'(\imath) = \{y' \in Y : \exists\, y \in \imath,\ \exists\, s \in \Sigma^* \ s.t.\ P_\Omega(s) = \epsilon \wedge\ y' = \theta(y, s)\},$$

- $y_0' = (UR'(x_0), UR'(\tilde{q}_0))$ is the initial states with $y_0 = (x_0, \tilde{q}_0)$.

By the definition of the transition function we have $y' = \theta'(y, \sigma) = UR'(NX_\sigma'(y)) = UR'(\theta(y, \sigma)) = \{y' \in Y : \exists\, s \in \Sigma^* \ s.t.\ P_\Omega(s) = P_\Omega(\sigma) \wedge y' = \theta(y, s)\}$. By extending the transition function to a string s in a usual way we get that $\theta'(y, s) = \{y' \in Y :\ \exists\, t \in \Sigma^* \ s.t.\ P_\Omega(s) = P_\Omega(t) \ \wedge\ y' = \theta(y, t)\} = \mathcal{E}_\Omega^{\mathbf{A} \times \mathbf{G}_{ag}}(s)$.

Next, we show that the state components of the observer's state reached upon $Obs(\mathbf{A} \times \mathbf{G}_{ag})$ always belongs to the state estimator value of the dynamic observer $Obs(\mathbf{G}_{ag})$ after an observable string which is available to the observer. For any $\imath \in Y'$, let

$$I_2(\imath) = \{\{\tilde{q}_1, \ldots, \tilde{q}_m\} \in 2^{\tilde{Q}} : \imath = (x_1, \ldots, x_k, \tilde{q}_1, \ldots, \tilde{q}_m)\}.$$

Proposition 1: Let $Obs(\mathbf{G}_{ag}) = (Z, \Sigma, \xi, Z_0)$ be the dynamic observer induced by (9), and s be an observable string available to the observer $Obs(\mathbf{A} \times \mathbf{G}_{ag}) = (Y', \Sigma, \theta', y_0')$. Then we have $I_2(\theta'(y_0', s)) \in \bigcup_{z_0 \in Z_0} \xi(z_0, s)$.

Finally, we formally show that the state space of the dynamic observer $Obs(\mathbf{A} \times \mathbf{G}_{ag})$ indeed yields the state estimate of the receiver. For $\imath \in 2^{\tilde{Q}}$, let

$$I_1(\imath) = \{q \in Q : \tilde{q} \in \imath\}.$$

Theorem 1: Let $\Omega = (\mathbf{A}, \mathcal{L})$ be an information transmission policy imposed on $\mathbf{G}$ and $s = \sigma_1\sigma_2 \ldots \sigma_m$ be an observable string available to the dynamic observer $Obs(\mathbf{A} \times \mathbf{G}_{ag}) = (Y', \Sigma, \theta', y_0')$. Then we have

$$I_1(I_2(\theta'(y_0', s))) = \mathcal{E}_\Omega^G(s).$$

Example 3: Let us consider system $\mathbf{G}_{ag}$ in Fig. 2. Given an information transmission policy $\Omega = (\mathbf{A}, \mathcal{L})$, where $\mathbf{A}$ is shown in Fig. 4 and $\mathcal{L}$ is given by: $l(x_0, \sigma_1) = l(x_0, \sigma_2) = l(x_3, \sigma_1) = l(x_3, \sigma_2) = l(x_4, \sigma_3) = l(x_4, \sigma_4) = N$ and $l(x_0, \sigma_3) = l(x_1, \sigma_2) = l(x_2, \sigma_1) = Y$. The corresponding product automata $\mathbf{A} \times \mathbf{G}_{ag}$ and the dynamic observer $Obs(\mathbf{A} \times \mathbf{G}_{ag})$ defined above are given in Fig. 4. Let string s be an arbitrary observable string available to the the dynamic observer $Obs(\mathbf{A} \times \mathbf{G}_{ag})$. It can be verified that $I_2(\theta'(y_0', s)) \in \bigcup_{z_0 \in Z_0} \xi(z_0, s)$ (Proposition 1) and $I_1(I_2(\theta'(y_0', s))) = \mathcal{E}_\Omega^G(s)$ (Theorem 1) always hold.

IV. Synthesis of Feasible Transmission Information Policies

In this section, we discuss how to synthesize a deterministic information policy that ensures the IS-based property is satisfied. Given an observer $Obs(\mathbf{G}_{ag}) = (Z, \Sigma, \xi, Z_0)$, we say that a state z is *consistent* if $\forall \sigma \in \Sigma_z \neq \emptyset$, $NX_\sigma(z) \neq \emptyset$. We denote by Z_{const} the set of consistent states in Z and we say Z is consistent if all reachable states are consistent.

Our approach for synthesizing a deterministic information policy consists of the following two steps: (i) first construct the largest sub-automata $\mathbf{G}^*$ of $Obs(\mathbf{G}_{ag})$ such that $L(\mathbf{G}^*)$ satisfies the IS-based property and the states of $\mathbf{G}^*$ are consistent; and (ii) then extract one deterministic information transmission policy Ω based on $\mathbf{G}^*$.

A. Synthesis of the transmission information policy

To satisfy the IS-based property φ, it should be guaranteed that, for any $\imath \in 2^{\tilde{Q}}$, $\varphi(\imath) = 1$.

To this end, we define

$$Z_{dis} = \{\imath \in Z : \varphi(\imath) = 0\}$$

as the set of states that dissatisfies the IS-based property φ.

In order to synthesize a desired transmission information policy, we firstly construct the largest sub-automata of $Obs(\mathbf{G}_{ag})(= (Z, \Sigma, \xi, Z_0))$ that enumerates all the feasible transitions satisfying the constraints of ξ. Such an all-feasible automaton is denoted by $\mathbf{G}_{total}$. Then, we need to delete some states that dissatisfy the IS-based property and obtain a new automaton $\mathbf{G}_0 = \mathbf{G}_{total} \upharpoonright_{Z \setminus Z_{dis}}$, where $\mathbf{G} \upharpoonright_Z$ denotes the automata obtained by restricting the state-space of $\mathbf{G}$ to Z.

However, by deleting partial states, the resulting automata may become inconsistent. Hence, we also need to delete inconsistent states recursively. To this end, we give an operator R that maps an automaton to a new one by: $R : \mathbf{G} \mapsto \mathbf{G} \upharpoonright_{Z_{const}}$ and define $\mathbf{G}^* = \lim_{k \to \infty} R^k(\mathbf{G}_0)$ as the largest consistent automaton which satisfies the IS-based property. An algorithm (Algorithm 1) is proposed to synthesize $\mathbf{G}^*$ via a depth-first search. By Algorithm 1, we will delete a state if there exists an event that are defined in $Obs(\mathbf{G}_{ag})$ and not defined at $R^k(\mathbf{G}_0)$ for some $k \to \infty$ (lines 3-5), namely, the deleted states are inconsistent. We obtain an automaton that satisfies the IS-based property by deleting all inconsistent states recursively.

Algorithm 1:

---

input : $Obs(\mathbf{G}_{ag}) = (Z, \Sigma, \xi, Z_0)$
output: $\mathbf{G}^* = (\tilde{Z}, \tilde{\Sigma}, \tilde{\xi}, \tilde{Z_0})$

```
1    G* = G_total ↾_{Z\Z_dis} = (Z̃, Σ, ξ̃, Z̃_0);
2    Z̃ = Z;
3    for each z ∈ Z do
4        if ∃σ ∈ Σ s.t. ξ(z,σ)! and ¬ξ̃(z,σ)! then
5            Z̃ = Z̃ \ {z};
     G* = G* ↾_Z̃ .
```

So far, the first step has been finished. Now we execute the second step by synthesizing an automata $\mathbf{A} = (\mathbf{X}, \mathbf{\Sigma_A}, \eta, \mathbf{x_0})$ from the largest feasible automaton $\mathbf{G}^*$ to present the corresponding information transmission policy, while ensures that the obtained automata $\mathbf{A}$ transmits as fewer events as possible. Specifically, we start from the initial state $q_0 \in Q$, check and choose the states labeled by N for each defined event as far as possible. We denote the set of states that contains $\tilde{q}$ in $Obs(\mathbf{G}_{ag})$ by $Q(\tilde{q}) = \{z \in Z | \tilde{q} \in z\}$. Then for each event σ defined at $Q(\tilde{q})$ in $Obs(\mathbf{G}_{ag})$, $Q(\tilde{q})$ can be divided into three sets $Q^Y_\sigma(\tilde{q})$, $Q^N_\sigma(\tilde{q})$, and $Q^u_\sigma(\tilde{q})$, where $Q^Y_\sigma(\tilde{q}) = \{z \in Q(\tilde{q}) | l(\tilde{q}, \sigma) = Y\}$ denotes the set of states that σ will be transmitted at state $\tilde{q}$, $Q^N_\sigma(\tilde{q}) = \{z \in Q(\tilde{q}) | l(\tilde{q}, \sigma) = N\}$ denotes the set of states that σ will not be transmitted at state $\tilde{q}$, and $Q^u_\sigma(\tilde{q}) = Q(\tilde{q}) \setminus (Q^Y_\sigma(\tilde{q}) \cup Q^N_\sigma(\tilde{q}))$ denotes the set of states that σ will be transmitted at some loops and not be transmitted at other loops. Then by choosing different transmission decision from $Q^Y_\sigma(\tilde{q})$, $Q^N_\sigma(\tilde{q})$, and $Q^u_\sigma(\tilde{q})$, for each state $\tilde{q}$ and σ we obtain a unique deterministic transmission decision. That is why we will obtain a deterministic transmission information policy.

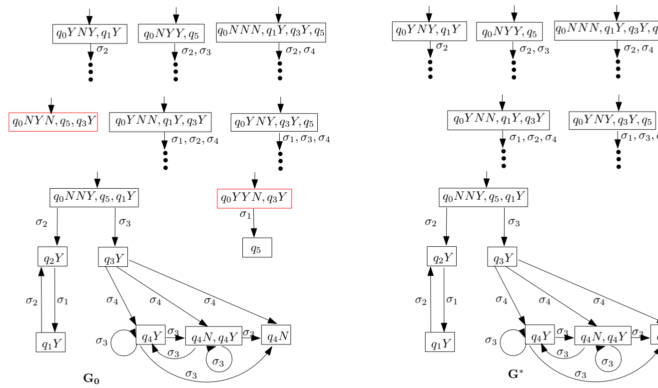

Fig. 5. $\mathbf{G}_0$ and $\mathbf{G}^*$

Example 4: Let us consider $Obs(\mathbf{G}_{ag})$ in Fig. 2. We firstly delete states that make the system $\mathbf{G}$ undistinguishable and obtain a distinguishable automata $\mathbf{G}_0$ as shown in Fig. 5. We note that there exist some states that are inconsistent in $\mathbf{G}_0$, e.g., (q_0NYN, q_3Y, q_5) and (q_0YYN, q_3Y). Event σ_2 is defined at states (q_0NYN, q_3Y, q_5) and (q_0YYN, q_3Y) in $Obs(\mathbf{G}_{ag})$ and not defined in $\mathbf{G}_0$. We thus need to delete inconsistent states recursively by Algorithm 1 and obtain the largest consistent and distinguishable automata $\mathbf{G}^*$ which is shown in Fig. 5. Next, we synthesize an automata $\mathbf{A}$ from the largest feasible automata $\mathbf{G}^*$, where the obtained automata $\mathbf{A}$ transmits events as few as possible by choosing the states labeled by N for each defined event. For instance, $\mathbf{G}^*$ contains q_4Y, (q_4Y, q_4N), and q_4N for the state $\tilde{q}_4$, to transmit fewer events here we keep q_4N and delete q_4Y and (q_4Y, q_4N). The resulting automata $\mathbf{A}$ is shown in Fig. 6.

B. Realization of the information transmission policy

In the last subsection, we synthesize an automata $\mathbf{A}$ from the largest feasible automata $\mathbf{G}^*$ to present a specific information transmission policy, where the labels of each event are given as the effect of $\mathbf{A}$. However, we can't accurately know how the information is transmitted at each step. There may exist some state-transition loops in $\mathbf{G}$ such that the label of the event in this loop with different history may be different. In order to accurately know how the information are transmitted at each step under a given policy $\mathbf{A} = (\mathbf{X}, \mathbf{\Sigma}, \eta, \mathbf{X_0})$, namely, obtain the labeling

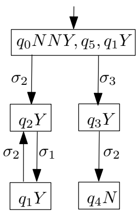

Fig. 6. Synthesized information transmission policy $\mathbf{A}$

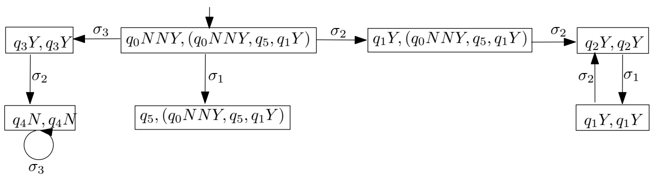

Fig. 7. The product automata $\mathbf{S} = \mathbf{G}_{ag} \times \mathbf{A}$

function $\mathcal{L}$, we construct a new non-deterministic automata $\mathbf{S} = \mathbf{G}_{ag} \times \mathbf{A} = (\mathbf{H}, \mathbf{\Sigma}, \zeta, \mathbf{h_0})$, where

- $H \subseteq \tilde{Q} \times X$ is the set of states,
- Σ is the set of events,
- $\zeta : H \times \Sigma \to H$ is the partial transition function defined by: for any $h = (\tilde{q}, x),\ h' = (\tilde{q}', x') \in H, \sigma \in \Sigma$, we have $h' \in \zeta(h, \sigma)$ iff

$$x' = \begin{cases} x, \text{ if } l(\sigma) = N \\ \eta(x, \sigma), \text{ if } l(\sigma) = Y; \end{cases}$$

$$q' = \delta(q, \sigma) \ \wedge \ \tilde{q}' \in \{q' \times \{Y, N\}^{|\Sigma_{q'}|}\} \cap x',$$

- $h_0 = (\tilde{q}_0, x_0)$ with $\tilde{q}_0 \in \{q_0 \times \{Y, N\}^{|\Sigma_{q_0}|}\} \cap x_0$ is the initial state.

After attaching the policy $\mathbf{A}$ to the system, the transmission label of each event at each step can be obtained by the first component of the states in $\mathbf{S}$ and is deterministic (even if $\mathbf{S}$ may be nondeterministic). For any state $h = (\tilde{q}, x) \in H$ and $\sigma \in \Sigma_h$, we have $\mathcal{L}(\tilde{q}, \sigma) = l(\sigma)$.

Example 5: Let us again consider $\mathbf{G}_{ag}$ in Fig. 2 and the information transmission policy $\mathbf{A}$ in Fig. 6. We attach the policy $\mathbf{A}$ to the system by the above definition and the obtained $\mathbf{S} = \mathbf{G}_{ag} \times \mathbf{A}$ is shown in Fig. 7.

Remark 1: We note that $\mathbf{S}$ may be non-deterministic, which is caused by transitions leading to the exact state estimate of the receiver while with different labels. To convert $\mathbf{S}$ to a deterministic automaton, we introduce a *relabeling* map $R : (H \times \Sigma \to H) \to (H \times (\Sigma \times \{Y, N\}^{|\Sigma|}) \to H)$ such that for any $h,\ h' \in H, \sigma \in \Sigma$,

$$R : (h \xrightarrow{\sigma} h') \to (h \xrightarrow{(\sigma, l_{h'})} h'),$$

where $l_{h'} = l(I_1(h'))$. We add a label to each transition, where the label is consistent with the label of the reachable state of this transition. By the relabeling map R, we obtain a new deterministic automaton $\mathbf{S}' = (H, \Sigma, \zeta', h_0)$, where $\zeta' : (H \times \Sigma \to H) \to (H \times (\Sigma \times \{Y, N\}^{|\Sigma|}) \to H)$ is defined by $\zeta'(h, (\sigma, l_{h'})) = h'$ iff $\zeta(h, \sigma) = h'$.
$\mathbf{G}_{ag} \times \mathbf{A}$ belongs to that of $Obs(\mathbf{G}_{ag})$ and thus gives the

By the above steps, we know that the information policy $\mathbf{A}$ is obtained by choosing a unique transmission decision for each event in $\mathbf{G}^*$ and $\mathbf{G}^*$ is obtained by deleting the bad states of $Obs(\mathbf{G}_{ag})$. Hence, the state space of $\mathbf{S} =$ state estimation of the receiver under $\mathbf{A}$. Therefore, the state space of $\mathbf{S}$ is consist with the state estimation of the receiver $(\mathcal{E}^G_\Omega(s))$.

V. Conclusion

In this paper, we studied the problem of optimal sensor scheduling for remote state estimation of discrete-event systems. We investigated a mechanism that selectively transmits the observable events according to an information transmission policy. We proposed algorithms for synthesizing deterministic information policies that ensure the IS-based property. In the future, we aim to consider the distributed and decentralized sensing and information transmission architecture based on the results proposed in this paper.

References

[1] Y. Liu, K. Cai, and Z. Li., "On scalable supervisory control of multi-agent discrete-event systems," *Automatica*, p. 108, 2019.

[2] Z. Ma and K. Cai, "Optimal secret protections in discrete-event systems," *IEEE Transactions on Automatic Control*, 2021.

[3] X. Yu, X. Yin, S. Li, and Z. Li, "Security-preserving multi-agent coordination for complex temporal logic tasks," *Control Engineering Practice*, vol. 123, p. 105130, 2022.

[4] G. M. Lipsa and N. C. Martins, "Remote state estimation with communication costs for first-order lti systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, p. 2013–2025, 2011.

[5] S. Trimpe and R. D'Andrea, "Event-based state estimation with variance-based triggering," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, p. 3266–3281, 2014.

[6] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete-event system," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 957–975, 2003.

[7] F. Lin, "Control of networked discrete event systems: dealing with communication delays and losses," *SIAM Journal on Control and Optimization*, vol. 52, no. 2, pp. 1276–1298, 2014.

[8] Y. Ji, X. Yin, and S. Lafortune, "Optimal supervisory control with mean payoff objectives and under partial observation," *Automatica*, vol. 123, p. 109359, 2021.

[9] Z. Ma, X. Yin, and Z. Li, "Marking diagnosability verification in labeled petri nets," *Automatica*, vol. 131, p. 109713, 2021.

[10] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, 2016.

[11] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, "Optimal sensor activation for diagnosing discrete event systems," *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.

[12] E. Dallal and S. Lafortune, "On most permissive observers in dynamic sensor activation problems observations," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, p. 966–981, 2014.

[13] W. Wang, C. Gong, and D. Wang, "Optimizing sensor activation in a language domain for fault diagnosis," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 743–750, 2018.

[14] X. Yin and S. Lafortune, "Minimization of sensor activation in decentralized discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3705–3718, 2018.

[15] X. Yin and S. Lafortune, "A general approach for optimizing dynamic sensor activation for discrete event systems," *Automatica*, vol. 105, pp. 376–383, 2019.

[16] N. Ran, J. Hao, and C. Seatzu, "Prognosability analysis and enforcement of bounded labeled petri nets," *IEEE Transactions on Automatic Control*, 2021.

[17] B. Zhang, S. Shu, and F. Lin, "Maximum information release while ensuring opacity in discrete event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 1067–1079, 2015.

[18] B. Behinaein, F. Lin, and K. Rudie, "Optimal information release for mixed opacity in discrete-event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1960–1970, 2019.