State Estimation for Timed Discrete Event Systems with Communication Delays

Chengshi Miao[1], Shaolong Shu[1], Feng Lin[1,2]
[1]School of Electronic and Information Engineering
Tongji University, Shanghai, 201804, China
[2]Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202, USA
(Email: miaochengshi@hotmail.com, shushaolong@hotmail.com, flin@wayne.edu)

***Abstract*—In this paper, we investigate state estimation for timed discrete event systems with communication delays. The problem is to estimate the current state information under the uncertainties caused by partial observations and random observation delays. We construct an augmented automaton by analyzing the influence mechanism of random observation delays exist between the supervisor and the modeled timed discrete event systems. The augmented automation shows all the possible observation trajectories and the current observation delay for each states. With these essential information, we extend the augmented automaton to a delayed observer and proposed a state estimation algorithm to achieve the correct state estimation results.**

***Index Terms*—Timed Model, Discrete Event Systems, Networked Control, State Estimation**

I. Introduction

Timed discrete event system [1] is a concept generalized from the discrete event system, in which lower and upper time bounds of events are specified and events must occur within their lower and upper time bound [2]. Besides, in timed discrete event system, time is specified as the time event t. Since communication networks are now widely used as information transmission medias in engineering systems, investigations on networked timed discrete event systems is of great importance. In a typical networked system, closed loop control is often implemented via a shared communication network, especially in discrete event systems which are essentially event driven [3][4]. In such networked discrete event systems, the supervisor and the controlled system, the supervisor observes the system through the observation channel, and sends control commend through the control channel. Due to the inherent property of communication networks, the communication delays in both observation channel and control channel are inevitable. Hence how to handle communication delays is a significant issue. Meanwhile, based on the model of timed discrete event systems, communication delays are measured by timed event t Thus the supervisory control problem of networked timed discrete event systems is more difficult. Naturally, the state estimation, which is the fundamental of most supervisory control problems, is of great significance in discrete event systems, especially when timed event and communication networks are both introduced. In this paper, we will systematically investigate state estimation for timed discrete event systems with communication delays.

State estimation is a key in many control engineering applications involving complex systems [6] and it is needed in may discrete event system theories such as supervisory control [7], fault diagnosis [8] and opacity [6]. For un-timed discrete event systems, state estimation is an important research topic and many theorems have been proposed. The earliest work on state estimation is done by in [10] In this work, the author investigates the ability to determine the current states of a discrete event system, which is very useful in the theoretic analysis of a number of basic supervisory control problems. Later, state estimation problems are investigated by other researchers [11][12][6]. In [11], the authors address the problem of state estimation in discrete event systems modeled by labeled Petri nets that may have nondeterministic transitions or unobservable transitions. The results show that the state estimation problem can be solved with complexity that is polynomial in the length of the observation sequence. In [12], a distributed state estimation algorithm is proposed to estimate the unavailable current state information. In the proposed algorithm, local sites maintain and update local state estimates based on their local observations of the plant behaviors, when there exist delays in observations. The algorithm ensures that the state estimates at all the local sites are optimal. The problem of decentralized state estimation is investigated in [6]. The authors consider a discrete event system modeled by a nondeterministic finite automaton whose underlying activity is partially observed. The proposed recursive algorithm in this paper is used to fuse the information about different subsets of events and infers the possible current states of the given system. Recently, we systematically investigates the problems of networked discrete event systems. Communication delays and losses are considered in both observation channel and control channel [13], and the state estimation of networked discrete event systems is investigated. Considering communication networks between supervisor and system add more difficulties to state estimation, but it is essential to solve many supervisory control problems [14][15][16][17][18]. In the above mentioned works on networked discrete event systems with communication delays, the number of occurrence of events are used to measure the delays. Since the occurrence of events is not a good measure of time, the results obtained using untimed networked discrete event systems tend to be relatively conservative. Hence timed discrete event system is used in this paper, in which time event t is used to measure communication delays[19]. [19] considers both uncertainties due to partial observation and uncertainties due to communication delays and losses to achieve correct estimation. This paper extends the work of state estimation in [19]. To the best of our knowledge no other works on state estimation of timed discrete event systems with communication delays have been proposed.

In this paper, we assume that communications between the supervisor and the plant is via a shared networked, and communication delays exist in the observation channel, so observations may be delayed, we further assume that the delays are random and upper-bounded. Based on this assumption, we use timed discrete event system to model the plant and investigate state estimation problems under communication delays. First we analyze the mechanism of communication delays and build a mathematic model to describe all the possible observations under communication delays. We then define the new language generated by the delayed system. In order to illustrate the language and show the specific mechanism of communication delays, we construct an augmented automaton whose generated language is equal to the new language. The constructed augmented automaton covers all the possible observation trajectories and can well describe the observations with communication delays. Based on the augmented automaton, we develop a state estimation algorithm and check the correctness of the algorithm.

The rest of the paper is organized as follows. Section II introduces timed discrete event systems and define the mathematic model for delay observations. Section III states the state estimation problems and gives an simple example to show the difficulties brought by communication delays. In Section IV, we develop an augmented automaton and prove important propositions and theorems to lay a foundation for the next section. In Section V, we propose algorithms for solving the state estimation problem and give an example to show the process and results. We conclude the paper in section VI.

II. Timed Discrete Event Systems with Communication Delays

A timed discrete event system is described by a timed automaton as

$$\widetilde{G} = (Q, \widetilde{\Sigma}, \rho, q_0)$$

where Q is the set of states, $\widetilde{\Sigma} = \Sigma \cup \{t\}$ is the set of events with time event t (represents tick of the global clock) where Σ denotes the set of activity events, $\rho : Q \times \widetilde{\Sigma} \to Q$ is the (partial) transition function, q_0 is the initial state. The rules of state transition function is derived from the activity transition function that can be referred in details from [1]. In the usual way, we extend the transition to $\rho : Q \times \widetilde{\Sigma}^* \to Q$. In addition, we use $\rho(q, s)!$ to mean that the transition $\rho(q, s)$ is defined, then the closed behavior of $\widetilde{G}$ is $L(\widetilde{G}) = \{s \in \widetilde{\Sigma}^* : \rho(q_0, s)!\}$.

We assume that some events in Σ are observable while the other events are unobservable. The set of observable events are denoted by $\Sigma_o \subseteq \Sigma$ and the set of unobservable events are denoted by $\Sigma_{uo} = \Sigma - \Sigma_o$. Note that event t should always be observable. Hence for timed discrete events systems, the observable event set is $\widetilde{\Sigma}_o = \Sigma_o \cup \{t\}$ and $\widetilde{\Sigma}_{uo} = \Sigma_{uo}$. The set of all possible transitions is denoted by ρ as $\rho = \{(q, \sigma, q') : \rho(q, \sigma) = q'\}$. The set of observable transitions is denoted by ρ_o as $\rho_o = \{(q, \sigma, q') : \rho(q, \sigma) = q' \wedge \sigma \in \widetilde{\Sigma}_o\}$, and the set of unobservable transitions is denoted by ρ_{uo} as $\rho_{uo} = \{(q, \sigma, q') : \rho(q, \sigma) = q' \wedge \sigma \in \widetilde{\Sigma}_{uo}\}$. Because $t \in \widetilde{\Sigma}_o$, all the transitions labeled with t in $\widetilde{G}$ are observable. The observation is the natural projection $P : \widetilde{\Sigma}^* \to \widetilde{\Sigma}_o^*$ defined as

$$P(\varepsilon) = \varepsilon \qquad P(s\sigma) = \begin{cases} P(s)\sigma & \sigma \in \widetilde{\Sigma}_o \\ P(s) & \sigma \notin \widetilde{\Sigma}_o \end{cases}$$

For a language L, all the observation is then defined as $P(L) = \cup_{s \in L} P(s)$.

As discussed in [13], when we observe the occurrence of events via networks, communication delays are unavoidable. We assume delays are random and is upper-bounded by N ticks. We also assume delays do not change the order of events occurred. Because delays are random, even for the same string s, the observation becomes nondeterministic. Let us use $\Phi_D^N(s)$ to denote the set of all the possible observations. The following will discuss how to obtain $\Phi_D^N(s)$. We first consider the case with all the events are observable, that is, $\widetilde{\Sigma} = \widetilde{\Sigma}_o$.

For a string s, we use $|t|(s)$ to denote the number of time events in s. For example, for string $s_1 = tt\sigma_1$, $|t|(s_1) = 2$; for string $s_2 = t\sigma_1 tt$, $|t|(s_2) = 3$. We then define an operation $RMV^i(s)$ to remove a suffix string s' with the length $|t|(s') = i$ as

$$RMV^i(s) = \{s' : s = s's'' \wedge |t|(s'') = i\}$$

Even though for any string $s' \in RMV^i(s)$, the last occurred substring containing i time events is delayed, we still can observe the elapse of time and know i time events have occurred. Hence, the observed event string is not $s' \in RMV^i(s)$, but defined as

$$\Omega^i(s) = \{s't^j \in \widetilde{\Sigma}^* : s' \in RMV^j(s) \wedge j \leq i\}$$

where $t^j = \underbrace{tt \cdots t}_{j}$.

The strings $s't^j$ defined by $\Omega^i(s)$ imply the delay always occurs after the last event of string s'. However, the delay in practice can occur for any time during the lifespan of the string s' which results in more possible observations for the same string s. In order to find all these observations, we define an operation $DELAY^j(\cdot)$ as follows.

For a given string s, we first divide it into different substrings as

$$s = u_1 t u_2 t \ldots u_k t u_{k+1}$$

where u_i does not include any time event, that is, $|t|(u_i) = 0, i = 1, 2, \ldots, k+1$. Note that u_i may be empty strings.

For any substring $u_i t$, any event in u_i may be delayed for one tick such that the event is observed after the time event t. Let us rewritten as $u_i t = \sigma_1\sigma_2\ldots\sigma_j t, i = 1, 2, \ldots, k$. When events in u_i are delayed for at most one tick, all the possible observations are then obtained by moving t forwardly from one event to another as

$$\begin{aligned}&TMV(\sigma_1\sigma_2\ldots\sigma_j t)\\&=\{\sigma_1\sigma_2\ldots\sigma_j t, \sigma_1\sigma_2\ldots t\sigma_j,\\&\ldots, \sigma_1 t\sigma_2\ldots\sigma_j, t\sigma_1\sigma_2\ldots\sigma_j\}\end{aligned}$$

Then for all $s = u_i t u_2 t\cdots u_k t u_{k+1} \in L(\widetilde{G})$, $DELAY^1(s)$ is defined as

$$\begin{aligned}&DELAY^1(s)\\&=TMV(u_1 t)TMV(u_2 t)\ldots TMV(u_k t)u_{k+1}.\end{aligned}$$

We extend $DELAY^1(s)$ into the case of j ticks as

$$DELAY^j(s) = DELAY^1(DELAY^{j-1}(s))$$

Combining these two operations of $\Omega^i(\cdot)$ and $DELAY^j(\cdot)$, we obtain the set of all possible observations with the delay bounded by N as

$$\begin{aligned}\Theta^N(s) &= DELAY^N(s)\circ\Omega^N(s)\\&=\cup_{s'\in\Omega^N(s)} DELAY^N(s')\end{aligned}$$

For language $L(\widetilde{G})$, the set of all possible observations under the delay is given by

$$\Theta^N(L(\widetilde{G})) = \cup_{s\in L(\widetilde{G})}\Theta^N(s)$$

Now we extend observation to the general case in which some events are unobservable by natural projection as

$$\begin{aligned}\Phi_D^N(s) = P(\Theta^N(s)) &= P(DELAY^N\circ\Omega^N(s))\\&=P(\cup_{s'\in\Omega^N(s)} DELAY^N(s'))\end{aligned}$$

We further extend the definition $\Phi_D^N(\cdot)$ from string to language $L(\widetilde{G})$ as

$$\Phi_D^N(L(\widetilde{G})) = \cup_{s\in L(\widetilde{G})}\Phi_D^N(s)$$

III. Problem Statement

In discrete event systems, state estimates are very useful for applications such as supervisory control and fault diagnosis [17][20]. If there are no communication delays, we can construct an observer using the traditional method and then calculate the current state estimate for any occurred event sequence with the observer [21]. However, when there are communication delays, even for a given event sequence, there are multiple possible observations and calculating the current state estimate becomes difficult. In timed discrete event systems, the delay defined by time event t adds more difficulties to the state estimation problem.

Let us first formally define the current state estimate after observing $w \in \Phi_D^N(L(\widetilde{G}))$ as

$$\begin{aligned}&TE^N(w)\\&=\{q\in Q : (\exists s\in L(\widetilde{G}))w\in\Phi_D^N(s)\wedge\rho(q_0, s) = q\}\end{aligned}$$

The following example is then used to illustrate the difficulty of calculating the state estimate $TE^N(w)$ under communication delays.

Example 1: We consider the timed discrete event system $\widetilde{G}$ shown in Figure 1. Assume that all events are observable. That is, $\widetilde{\Sigma} = \widetilde{\Sigma}_o = \{\alpha, \beta, t\}$. Assume there is one tick delay, that is $N = 1$. Let us consider how to calculate the current state estimate for the observed event sequence string $w = t\alpha t$.

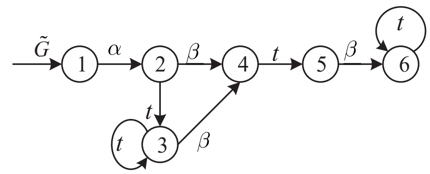

Fig. 1. A timed discrete event system $\widetilde{G}$

In order to calculate the current state estimate, we need to find all the possible occurred event sequences for the observed event sequence w. Because the observed event sequence includes two time events, the actually occurred event sequence must be one of those event sequences that includes two time events. We enumerate all such possible event sequences in $L(\widetilde{G})$. They are $\alpha\beta t\beta t, \alpha tt, \alpha tt\beta, \alpha t\beta t, \alpha t\beta t\beta$. Let us take event sequence $s = \alpha\beta t\beta t$ for example to show how to calculate the observations.

We have

$$RMV^0(s) = \{\alpha\beta t\beta t\}, RMV^1(s) = \{\alpha\beta t\beta, \alpha\beta t\}$$

We then have

$$\Omega^1(s) = \{\alpha\beta t\beta t, \alpha\beta tt\}$$

For string $\alpha\beta t\beta t \in \Omega^1(s)$, it can be divided into substrings as

$$u_1 t = \alpha\beta t, u_2 t = \beta t, u_3 = \varepsilon$$

We use operation TMV to acquire all the possible observations as

$$TMV(\alpha\beta t) = \{\alpha\beta t, \alpha t\beta, t\alpha\beta\}, TMV(\beta t) = \{\beta t, t\beta\}$$

and then

$$\begin{aligned}&DELAY^1(\alpha\beta t\beta t) = TMV(u_1 t)TMV(u_2 t)\\&=\{\alpha\beta t\beta t, \alpha\beta tt\beta, \alpha t\beta\beta t, \alpha t\beta t\beta, t\alpha\beta\beta t, t\alpha\beta t\beta\}\end{aligned}$$

In the same way, we calculate $DELAY^1(\alpha\beta tt)$ as

$$DELAY^1(\alpha\beta tt) = \{\alpha\beta tt, \alpha t\beta t, t\alpha\beta t\}$$

Then all the possible observations generated by the delay bounded by 1 tick is

$$\begin{aligned}\Theta^1(s) &= DELAY^1(s)\circ\Omega^1(s) = \cup_{s'\in\Omega^1(s)} DELAY^1(s')\\&=\{\alpha\beta t\beta t, \alpha\beta tt, \alpha\beta tt\beta, \alpha t\beta\beta t, \alpha t\beta t,\\&\alpha t\beta t\beta, t\alpha\beta\beta t, t\alpha\beta t, t\alpha\beta t\beta\}\end{aligned}$$

Finally, since we assume all the events are observable in this example, so we have

$$\Phi_D^N(s) = \Phi_D^1(s) = P(\Theta^1(s))$$
$$= \{\alpha\beta t\beta t, \alpha\beta tt, \alpha\beta tt\beta, \alpha t\beta\beta t, \alpha t\beta t,$$
$$\alpha t\beta t\beta, t\alpha\beta\beta t, t\alpha\beta t, t\alpha\beta t\beta\}$$

In the same way, we calculate the observations for all the other four strings to see if the actually observed string $s = t\alpha t$ is among the set of observations. The results are shown in Table 1.

From Table 1, we know that when $N = 1$, $w \notin \Phi_D^N(\alpha\beta t\beta t)$, $w \in \Phi_D^N(\alpha tt)$, $w \in \Phi_D^N(\alpha tt\beta)$, $w \in \Phi_D^N(\alpha t\beta t)$ and $w \in \Phi_D^N(\alpha t\beta t\beta)$. Hence the current state estimate after observing $w = t\alpha t$ is the set of states which can be reached by event sequences $\alpha tt, \alpha tt\beta, \alpha t\beta t, \alpha t\beta t\beta$. From the given automaton $\widetilde{G}$, we have $\rho(q_0, \alpha tt) = 3$, $\rho(q_0, \alpha tt\beta) = 4$, $\rho(q_0, \alpha t\beta t) = 5$, and $\rho(q_0, \alpha t\beta t\beta) = 6$. Hence the current state estimate is $TE^N(w) = \{3, 4, 5, 6\}$.

In Example 1, in order to determine the current state estimate, we need to enumerate all the possible occurred event sequences and calculate all the observations for all the the possible occurred event sequences. The computation is rather complex and becomes infeasible when the length of the given observation increases. In the following, we want to find an effective way to calculate the state estimate for a given observed event sequence. Formally, the state estimation problem is stated as follows.

State Estimation for Timed Discrete Event Systems with Communication Delays: Consider a given networked timed discrete event systems $\widetilde{G}$ with communication delays bounded by N ticks. For a given observation $w \in \Phi_D^N(L(\widetilde{G}))$, find a way to calculate its state estimate $TE^N(w)$.

IV. Augmented Automaton

In order to calculate the state estimate $TE^N(w)$ with the observed event sequence w, the first step is to find all the possible observations $\Theta^N(L(\widetilde{G}))$. Our method is to construct an augmented automaton of which the language is equal to $\Theta^N(L(\widetilde{G}))$ as

$$\widetilde{G}^N = (Q^N, \widetilde{\Sigma}, \rho^N, q_0^N) = Ac(Q \times \Delta, \widetilde{\Sigma}, \rho^N, q_0^N)$$

where $\Delta = \{0, 1, 2, \cdots, N\}$ and $Ac(\cdot)$ denotes the accessible part. Every state in Q^N is a pair of which the first element is the state that the system stays and the second element is the number of delays currently. Initially, there are no delays. Hence the initial state $q_0^N = (q_0, 0)$. The transitions of the augmented automation is defined as follows.

1. For any state $(q, n) \in Q^N$, $n \leq N$, and any event $\sigma \in \widetilde{\Sigma}$, we have:

$$\rho_1^N = \{((q, n), \sigma, (q', n)) : \rho(q, \sigma) = q'\} \tag{1}$$

2. For any state $(q, n) \in Q^N$, one more tick delay can occur if $n < N$, hence we have

$$\rho_2^N = \{((q, n), t, (q, n+1)) : q \in Q \wedge n < N\} \tag{2}$$

3. For any state $(q, n) \in Q^N$, if $\rho(q, t)!$, then the occurrence of time event t will reduce delay by one tick. That is,

$$\rho_3^N = \{((q, n), \varepsilon, (q', n-1)) : \rho(q, t) = q' \wedge n > 0\} \tag{3}$$

Then we can formally define the transitions ρ^N as

$$\rho^N = \rho_1^N \cup \rho_2^N \cup \rho_3^N$$

Note that $\widetilde{G}^N$ is nondeterministic. Now let us use an example to illustrate how to construct the augmented automaton $\widetilde{G}^N$.

Example 2: We still consider the timed discrete event system in Example 1 and assume the observation delays is bounded by $N = 1$ tick. Then according to the definition of the transition function ρ^N, we can construct the augmented automaton following step 1,2,3 in Equations (1),(2) and (3) as follows. In Figure 2, we use red transition, blue transition and green

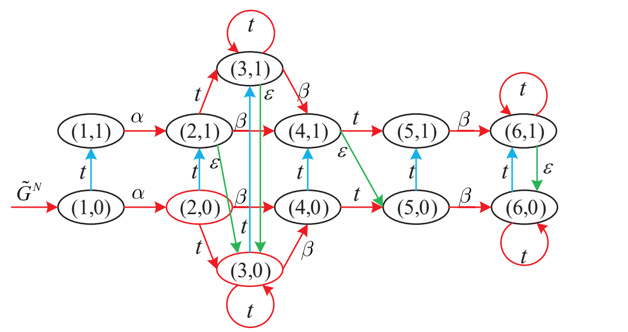

Fig. 2. Constructed automaton under the definition of transition function

transition to denote the transitions defined in Equations (1), (2) and (3) respectively.

The augmented automaton $\widetilde{G}^N$ has the following properties that we can use to calculate the current state estimation.

Propsition 1: For any trajectory tr generated by $\widetilde{G}^N$ such that

$$tr = (q_0, 0)\sigma_1(q_1, n_1)\sigma_2 \cdots \sigma_m(q_m, n_m)$$

we can find a string $s \in L(\widetilde{G})$ such that

$$(\exists s_R \in RMV^{n_m}(s))\sigma_1\sigma_2 \cdots \sigma_m \in DELAY^N(s_R t^{n_m})$$
$$\wedge q_m = \rho(q_0, s_R)$$

Propsition 2: For any $s \in L(\widetilde{G})$ and any substring with n_m delays: $s_R \in RMV^{n_m}(s)$, for any string $w \in DELAY^N(s_R t^{n_m})$, write

$$w = \sigma_1\sigma_2 \cdots \sigma_m$$

There must exist a trajectory tr generated by $\widetilde{G}^N$ such that

$$tr = (q_0, 0)\sigma_1(q_1, n_1)\sigma_2 \cdots \sigma_m(q_m, n_m) \wedge q_m = \rho(q_0, s_R)$$

From Proposition 1 and Proposition 2, we have the following theorem.

Theorem 1: The language generated by $\widetilde{G}^N$ equals to the language generated by Θ^N, that is,

$$L(\widetilde{G}^N) = \Theta^N(L(\widetilde{G}))$$

TABLE I
OBSERVATIONS OF EVENT SEQUENCES WITH $N = 1$

$s \in L(\widetilde{G})$	$\Phi_D^N(s)$	$w \in \Phi_D^N(s)$
$\alpha\beta t\beta t$	$\alpha\beta t\beta t, \alpha\beta tt, \alpha\beta tt\beta, \alpha t\beta\beta t, \alpha t\beta t, \alpha t\beta t\beta, t\alpha\beta\beta t, t\alpha\beta t, t\alpha\beta t\beta$	No
αtt	$\alpha tt, t\alpha t$	Yes
$\alpha tt\beta$	$\alpha tt, \alpha tt\beta, t\alpha t, t\alpha t\beta$	Yes
$\alpha t\beta t$	$\alpha t\beta t, \alpha tt, \alpha tt\beta, t\alpha\beta t, t\alpha t, t\alpha t\beta$	Yes
$\alpha t\beta t\beta$	$\alpha t\beta t, \alpha t\beta t\beta, \alpha tt, \alpha tt\beta, \alpha tt\beta\beta, t\alpha\beta t, t\alpha\beta t\beta, t\alpha t, t\alpha t\beta, t\alpha t\beta\beta$	Yes

Theorem 1 considers the situation when all events are observable. If we assume that some events are unobservable, we only need add a projection. Since for a string s, $P(\Theta^N(s)) = \Phi_D^N(s)$, and it can be extended from string to language, we have the following corollary.

Corollary 1: Under partial observation, we have

$$P(L(\widetilde{G}^N)) = \Phi_D^N(L(\widetilde{G}))$$

V. ALGORITHMS FOR STATE ESTIMATION

In this section, we investigate state estimation problem of timed discrete event systems with communication delays, and find a way to calculate the set of states systems may be in after observing $w \in \Phi_D^N(L(\widetilde{G}))$. We estimate system states based on the augmented automaton $\widetilde{G}^N$ constructed above. Since in a more general situation, not all events are observable, we will investigate state estimation problem under partial observation. We first replace all unobservable transitions in $\widetilde{G}^N$ by ε-transitions and denote the resulting automaton by $\widetilde{G}^N_\varepsilon$, so we have

$$\widetilde{G}^N_\varepsilon = \{Q^N, \widetilde{\Sigma}_o, \rho^N_\varepsilon, q_0^N\}$$

Obviously, $\widetilde{G}^N_\varepsilon$ is a non-deterministic automaton. We then convert it into an equivalent deterministic automaton, that is, the observer $\widetilde{G}^N_{obs}$ as an operator OBS on $\widetilde{G}^N_\varepsilon$.

$$\begin{aligned}\widetilde{G}^N_{obs} &= OBS(\widetilde{G}^N_\varepsilon)\\ &= (X, \widetilde{\Sigma}_o, \xi, x_0) = Ac(2^{(Q^N)}, \widetilde{\Sigma}_o, \xi, UR(q_0^N))\end{aligned}$$

where $Ac(\cdot)$ denotes the accessible part, and UR denotes the unobservable reach, defined for $x \subseteq Q^N$ as

$$\begin{aligned}UR(x) =&\{(q,n) \in Q^N\\ &: (\exists (q',n') \in x)\rho^N((q',n'),\varepsilon) = (q,n)\}\end{aligned}$$

The transition function ξ is defined for $x \in X$ and $\sigma \in \widetilde{\Sigma}_o$ as

$$\begin{aligned}\xi(x,\sigma) = UR(\{&(q,n) \in Q^N\\ &: (\exists (q',n') \in x)\rho^N((q',n'),\sigma) = (q,n)\})\end{aligned}$$

By the definition of the augmented states, for an observed string w that lead to augmented state (q,n), if $n > 0$, then the current observation delay is not zero. In other words, there exists delays in observation. The augmented states that contain non-zero current observation delays is not the states that system may be in after observing w. For a given observation w, let us see how to find its state estimation $TE(w)$.

For a set of possible augmented states $Q'^N \subseteq Q^N$, with the consideration of corresponding current observation delays, the set of states reachable from Q'^N is defined by a delayed reach operation DR as

$$\begin{aligned}DR(Q'^N) = \{(q,0) \in Q^N :&(\exists (q',n) \in Q'^N)(\exists s \in \widetilde{\Sigma}^*)\\ &|t|(s) = n \wedge \rho(q',s) = q\}\end{aligned}$$

To consider communication delays, we extend each states $x \in X$ to $DR(x)$ and use y to denote the resulting state. Y is the set of all the extensions. That is to say, if $X = \{x_1, x_2, \cdots, x_m\}$, then $Y = \{y_1, y_2, \cdots, y_m\}$, and for every $x_i \in X$, the corresponding y_i is given by $y_i = DR(x_i)$. Then we define a new observer called the delayed observer as

$$\widetilde{G}^N_{D,obs} = (Y, \widetilde{\Sigma}_o, \zeta, y_0)$$

where $Y \subseteq 2^{(Q^N)}$ contains all the set of states in which the current observations delays equal to zero. The transition function is defined for $y_i, y_j \subseteq Y$ and $\sigma \in \widetilde{\Sigma}_o$ as $\zeta = \{(y_i,\sigma,y_j) : (x_i,\sigma,x_j) \in \xi\}$.

With the above discussion, we conclude an algorithm for constructing the delayed observer in Algorithm 1.

Algorithm 1 Calculating delayed observer $\widetilde{G}^N_{D,obs}$

Input: $\widetilde{G}, N$
Output: $\widetilde{G}^N_{D,obs}$

1: Construct $\widetilde{G}^N = \{Q^N, \widetilde{\Sigma}, \rho^N, q_0^N\}$ based on $\widetilde{G}$ and N.
The process of constructing should obey Equations (1),(2),(3).
2: Replace all $\sigma \in \widetilde{\Sigma}_{uo}$ by ε to construct $\widetilde{G}^N_\varepsilon$.
After this step, we have $\widetilde{G}^N_\varepsilon = \{Q^N, \widetilde{\Sigma}_o, \rho^N_\varepsilon, q_0^N\}$.
3: Construct the observer $\widetilde{G}^N_{obs} = \{X, \widetilde{\Sigma}_o, \xi, x_0\}$ of $\widetilde{G}^N_\varepsilon$.
In this step, the initial states x_0 is given by $x_0 = UR(q_0^N)$, and the transition function ξ is defined for all $x \in X$ as $\xi(x,\sigma) = UR(\{(q,n) \in Q^N : (\exists (q',n') \in x)\rho^N((q',n'),\sigma) = (q,n)\})$.
4: Extend all states (including initial state) in $\widetilde{G}^N_{obs}$ by delayed reach.
For each states $x_i \in X$, extend it to y_i as $y_i = DR(x_i)$
5: Extend all transitions in $\widetilde{G}^N_{obs}$ to $\widetilde{G}^N_{D,obs}$, denote the new transitions by ζ.
For all $x_i, x_j \in X$ and $y_i, y_j \in Y$, $\zeta = \{(y_i,\sigma,y_j) : (x_i,\sigma,x_j) \in \xi\}$.
6: Construct delayed observer $\widetilde{G}^N_{D,obs} = \{Y, \widetilde{\Sigma}_o, \zeta, y_0\}$.
7: **return** $\widetilde{G}^N_{D,obs}$.
End

The following theorem shows that the delayed observer calculates the state estimate for a given observation under communication delays bounded by N ticks.

Theorem 2: With communication delays bounded by N ticks, for a given observation w, the state estimate $TE^N(w)$ is acquired by

$$TE^N(w) = ET(\zeta(y_0, w))$$

where $ET(\cdot)$ is defined to extract all the first elements in these states to get a new set as

$$ET(Q'^N) = \{q : (q, n) \in Q'^N\}$$

Example 3: Consider a timed discrete event system $\widetilde{G}$ as in Example 1. Suppose all the events are observable and the communication delays are bounded by 1 tick. First we follow the steps in Algorithm 1 to construct the delayed observer $\widetilde{G}^N_{D,obs}$. Based on the constructed augmented automaton $\widetilde{G}^N$ in Figure 2, we construct the observer $\widetilde{G}^N_{obs}$ as shown in Figure 3.

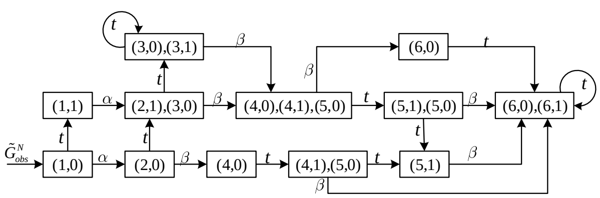

Fig. 3. The constructed observer $\widetilde{G}^N_{obs}$

In this figure, we see that $\widetilde{G}^N_{obs}$ is a deterministic automaton. Then based on this observer, we follow the steps in Algorithm 1 to construct the delayed observer $\widetilde{G}^N_{D,obs}$, we show this automaton in Figure 4.

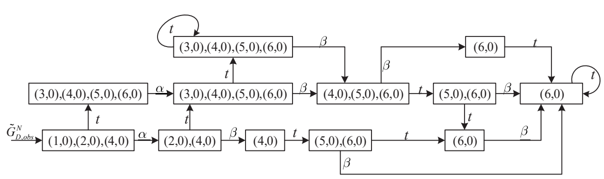

Fig. 4. The constructed delayed observer $\widetilde{G}^N_{D,obs}$

With this delayed observer, we can calculate the state estimation of $w = t\alpha t$ by Theorem 2. First we calculate $\zeta(y_0, w) = \{(3,0), (4,0), (5,0), (6,0)\}$, and by the definition of the operation $ET(\cdot)$, we have $ET(\zeta(y_0, w)) = \{3, 4, 5, 6\}$. By Theorem 2, the state estimation after observing w is

$$TE^N(w) = ET(\zeta(y_0, w)) = \{3, 4, 5, 6\}$$

It is the same as the one obtained in Example 1.

VI. Conclusion

In this paper, we systematically investigate the state estimation of timed discrete event systems with communication delays. The main contributions of this paper are as follows. (1) We specifically investigate the mechanism of communication delays in observations and define a new language. (2) We construct an augmented automaton with current observation delay to illustrate all the possible observed trajectories. (3) We present an effective algorithm for state estimation under partial observation and communication delays.

In the future, we will introduce control delays and communication losses in the analysis, and investigate supervisory control problems of timed discrete event systems with communication delays and losses.

References

[1] B. A. Brandin, W. M. Wonham, Supervisory Control of Timed Discrete-Event Systems. IEEE Transactions on Automatic Control, 39(2), pp.329-342, 1994.

[2] F. Lin, W. M. Wonham, Supervisory Control of Timed Discrete-Event Systems Under Partial Observation. IEEE Transactions on Automatic Control, 40(3), pp.558-562, 1995.

[3] H. V. Dimitrios, S. L. William, Handbook of Networked and Embedded Control Systems. Birkhauser, Boston, 2015.

[4] J. Lunze. Control Theory of Digitally Networked Dynamic Systems. Springer, 2014.

[5] S. Balemi, Input/Output Discrete Event Processes and Communication Delays. Discrete Event Dynamic Systems, 4(1), pp.41-85, 1994.

[6] C. N. Hadjicostis and C. Seatzu, Decentralized state estimation in discrete event systems under partially ordered observation sequences, 2016 13th International Workshop on Discrete Event Systems, pp.367-372, 2016.

[7] P. J. Ramadge, W. M. Wonham, Supervisory control of a class of discrete event processes, SIAM Journal on Control and Optimization, 25(1), pp.206C230, 1987.

[8] F. Basile, M. P. Cabasino, C. Seatzu, State Estimation and Fault Diagnosis of Labeled Time Petri Net Systems With Unobservable Transitions, IEEE Transactions on Automatic Control, 60(4), pp.997-1009, 2015.

[9] A. Saboori, C. N. Hadjicostis, Verification of Infinite-Step Opacity and Complexity Considerations, IEEE Transactions on Automatic Control, 57(5), pp.1265-1269, 2012.

[10] P. J. Ramadge, Observability of discrete event systems, 25th IEEE Conference on Decision and Control, pp.1108-1112, 1986.

[11] Y. Ru, C. N. Hadjicostis, State Estimation in Discrete Event Systems Modeled by Labeled Petri Nets, Proceedings of the 45th IEEE Conference on Decision and Control, pp.6022-6027, 2006.

[12] S. Xu, R. Kumar, Distributed state estimation in discrete event systems, 2009 American Control Conference, pp.4735-4740, 2009.

[13] F. Lin, Control of Networked Discrete Event Systems: Dealing with Communication Delays and Losses. SIAM J. Control Optim., 52(2), pp:1276-1298, 2014.

[14] S. Shu, F. Lin, A Predictive Approach for Networked Control of Discrete Event Systems. 28th Chinese Control and Decision Conference, pp.5482-5487, 2016.

[15] F. Wang, S. Shu, F. Lin, On Network Observability of Discrete Event Systems. 54th IEEE Conference on Decision and Control, pp.3528-3533, 2015.

[16] F. Wang, S. Shu, F. Lin, Robust Networked Control of Discrete Event Systems. IEEE Transactions on Automation Science and Engineering, 13(4), pp.1528-1540, 2016.

[17] S. Shu, F. Lin, Supervisor Synthesis of Networked Discrete Event Systems with Communicaton Delays. IEEE Transactions on Automatic Control, 60(8), pp.2183-2188, 2015.

[18] S. Shu, F. Lin, Decentralized Control of Networked Discrete Event Systems with Communication Delays. Automatica, 50(8), pp.2108-2112, 2014.

[19] B. Zhao, F. Lin, C. Wang, X. Zhang, M. P. Polis, L. Y. Wang, Supervisory Control of Networked Timed Discrete Event Systems and its Applications to Power Distribution Networks. IEEE Transactions on Control of Network Systems, to appear.

[20] M. Sampath, R. Sengupta, S. Lafortune, et al., Diagnosability of Discret Event Syatems. IEEE Transactions on Automatic Control, 40(9), pp.1555-1575, 1995.

[21] C. G. Cassandras, S. Lafortune, Introduction to Discrete Event Systems. Springer, Berlin, 2008.