Responses to the comments on the paper

Over-Approximation State Estimation for Networked Timed Discrete Event Systems with Communication Delays and Losses (Paper ID: SMCA-23-07-2013)

Zhaoyu Xiang, Yufeng Chen, Naiqi Wu, and Zhiwu Li

The comments of the editor and reviewers are in *italic*. The authors' response is regular and colored blue.

****

Editor-In-Chief:

Dear Dr Zhiwu Li:

Based on the referee reports and the Associate Editors recommendation with which I concur, I regret to inform you that your paper, in its current form, cannot be accepted for publication in the Transactions. Your paper does have merit and you are welcome to revise the manuscript keeping in mind the reviewer comments and to resubmit your paper as a new submission.

When submitting the revised version, please include a detailed statement containing the Authors' Responses to a Reviewer's Comments to address each reviewer's comments. If the authors do not agree with a comment, clearly explain the reason in the Authors' Responses.

Thank you for submitting to the IEEE Transactions on Systems, Man and Cybernetics and we hope to be able to review your paper again.

Sincerely,

Robert Kozma, FIEEE, FINNS Editor-In-Chief

Dear Editor-in-Chief Robert Kozma:

Thank you for your careful reading and giving us the opportunity to revise the manuscript entitled 'Over-Approximation State Estimation for Networked Timed Discrete Event Systems with Communication Delays and Losses.' We appreciate the time and efforts that the editor and reviewers have devoted to our manuscript. The insightful comments with high quality are immensely helpful and enable us to greatly improve the quality and presentation of the paper. We have addressed all comments and suggestions provided by the editor and reviewers in the revised version of our manuscript.

We hope that the revised manuscript will meet the demanding requirements of IEEE Transactions on Systems, Man, and Cybernetics: Systems, and we are happy to consider further revisions. We thank you again for your continued interest in our research and look forward to hearing from you at your earliest convenience.

Editor Comments for Authors:

Thanks for submitting the paper to TSMC. The paper was reviewed and, based on the comments, the paper cannot be accepted. The reviewers pointed out some concerns, which are summarized as follows: The motivation of the proposed work is unclear. The technical intricacies need to be described in an easy to read, easy to follow way. Other questions include multiple control channel, control loss not included, the difficulties of multiple channels and solutions, pros/cons of without setting FIFO, unclear statements and errors in description, hard-to-follow proof.

Other concerns include too many notations (a table to clarify them is needed), and presentation needs to be improved.

Response: We thank the editor for giving us an opportunity to revise the paper. We have addressed all the concerns raised and made significant improvements to the paper.

Specifically, we have clarified the motivation behind this work and simplified the technical intricacies to enhance readability and comprehension. Additionally, we have provided a detailed explanation for our choice of using a single control channel, emphasizing its commonality in the field as well as its research and application value. To support this, we have referenced results that are documented in the literature and presented an example from an intelligent traffic light system.

Furthermore, we have incorporated the scenario of control loss, along with corresponding discussions and specific methods for addressing it. We have also delved into the challenges and solutions associated with multiple channels, outlined the advantages and disadvantages of not setting FIFO, rectified unclear statements and errors in the description, and revised the proof for clarity. Moreover, we have included a table to clarify the numerous notations used in the paper. We have also carefully revised the presentation throughout the manuscript, striving for clearer and more coherent languages.

We hope that the revised manuscript will meet the demanding requirements of IEEE Transactions on Systems, Man, and Cybernetics: Systems, and we are happy to consider further revisions.

****

Reviewer's Comments to Authors:

Reviewer 1:

What are the contributions of the paper:

This paper studies the state estimation for a networked timed discrete event system, where a plant is controlled by a supervisor through a communication network composed of multiple channels with bounded communication delays and intermittent observation losses.

Response: We appreciate the comments of the reviewer.

What are the additional ways in which the paper could be improved:

1) The technical intricacies of the paper are challenging to navigate, and the interpretations are insufficient. This renders the paper to be comprehended, potentially limiting its accessibility to readers. Consequently, I'm worried that the readers might not obtain strong interest within this context.

Response: We thank the reviewer's comments. To address the concerns raised, we have made significant revisions to the paper, aiming to simplify the technical intricacies and expand the interpretations.

First, recognizing the challenge posed by the complex technical details, we have clarified the technical framework and outlined the problem-solving approach more explicitly. Specifically, we have provided a clear explanation of how we address communication delays and losses in observation and control channels within the context of a networked timed discrete event system (NTDES). We have augmented a plant and its supervisor by considering the dynamics of communication channels to effectively handle communication delays and losses. Based on this augmentation, we have constructed a compensated system, as depicted in Fig. 1, that offers an over-approximation of the actual closed-loop system's behavior. This technical framework is emphasized in the abstract of the paper, facilitating easier comprehension and navigation for readers.

Second, to further aid comprehension, we have included multiple figures in the paper to provide more visual explanations of important technical details. We have also provided a comprehensive Table I on Page 2, summarizing all notations used in the paper. This allows readers to quickly capture and understand each symbol, reducing confusion and enhancing clarity.

Third, we have significantly expanded the discussions in the paper to offer deeper insights into our methodology and its practical applications. On Page 7, Remark 1 and Example 3 now provide

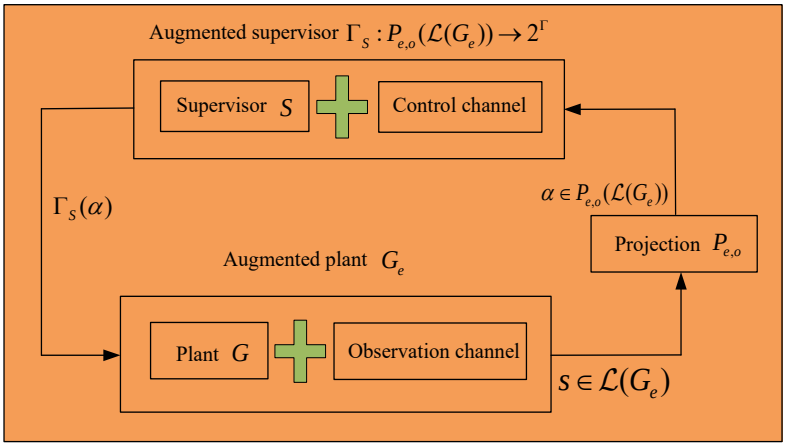

Figure 1: The compensated system $\Gamma_S/\mathfrak{N}$.

more detailed explanations of how the compensated system effectively handles control delays and losses. Moreover, Remark 1 not only indicates the applicability of the proposed method to first-in-first-out (FIFO) control channels but also provides a detailed explanation of its application to non-FIFO control channels. This discussion offers a deeper understanding, demonstrating the method's broad applicability across different types of control channels. Furthermore, in the conclusion section on Page 12, we explicitly compare and interpret the advantages and disadvantages of our approach with those of similar methods, presenting a comprehensive overview of its versatility and limitations.

Finally, to demonstrate the effectiveness of the proposed method, we have provided a simulation example accessible via `https://github.com/lahe1/SMC_simulation`. The technical specifications and results of the simulation are detailed in Appendix A.2 of this response letter. Such a comprehensive simulation is rare among similar works and we believe that its comprehensiveness will significantly pique readers' interest and encourage them to explore the paper further.

In summary, we hope that these revisions have adequately addressed the reviewer's concerns regarding technical intricacies and interpretations. The paper now features a clearer and more accessible technical framework, enriched discussions, and a practical simulation example that illustrates the effectiveness of the proposed method. We hope these efforts will foster a deeper understanding of our work and spark interest among readers.

2) This work considers multiple observation channels but does not consider multiple control channels. Why? And what are its benefits?

Response: Thank you for your insightful question. We choose to focus on a single control channel in this work primarily due to two reasons. First, the complexity of the research already poses significant challenges. Hence, incorporating multiple control channels would add further layers of complexity to the research. This would make it difficult to provide an adequate and comprehensive description of the proposed method while adhering to the specified page limit. Second, restricting our attention to a single control channel aligns with a common practice in research within this field, as exemplified by the studies reported in [4]–[8], [13]–[18], [23], [24].

The benefits of this approach are as follows. First, it allows for a more focused analysis of the system's design, taking into account both observation and control delays and losses. Second, even with a single control channel, the proposed NTDES $\mathfrak{N}$ has potential applications, e.g., in modeling an intelligent traffic light system, as depicted in Fig. 2.

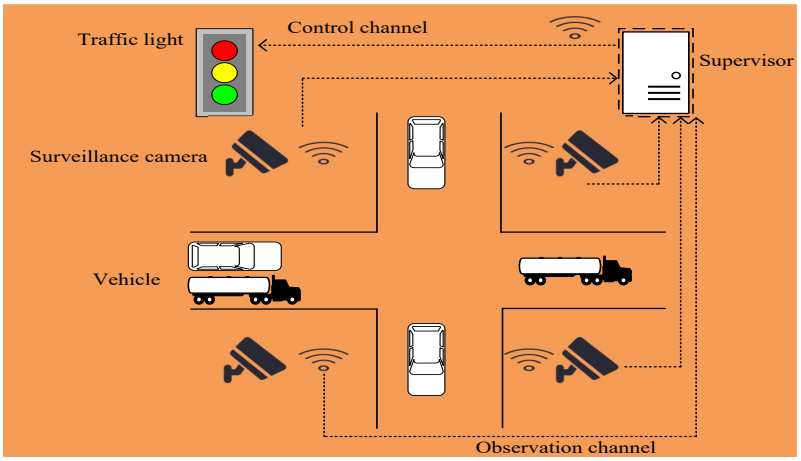

Figure 2: An intelligent traffic light system.

When specific components such as cameras are damaged, the system's state estimation capabilities become particularly crucial to predict traffic flow. As control signals may encounter delays or losses due to network fluctuations, robust state estimation methods are essential. This underscores the significance of studying systems with a single control channel, not only for their theoretical value but also for their potential in real-world applications. Looking ahead, we consider extending our work to multiple control channels.

3) As claimed in the Introduction "communication delays (resp. losses) include the observation delays (resp. losses) in observation channels and the control delays (resp. losses) in control channels." Why control losses do not be modeled in Definition 1?

Response: Thank you for pointing out this oversight. In the initial Definition 1, we unintentionally omitted modeling control losses. We have now revised Definition 1 on Page 3 of the paper to include the upper bound on control delays and losses within the control channel, denoted by $cc \in \mathbb{N}$.

To comprehensively handle control delays and losses, we have further modified Assumption A2 on the same page as follows: "A plant operates using the most recently received control decision in case a new decision issued by a supervisor experiences delay or loss in the control channel. No control decision remains delayed for more than cc time units after its issuance, and the plant must receive at least one control decision within every cc time unit period." Under this assumption, the plant always has a valid control decision to operate with, despite potential delays or losses in the control channel. Specifically, the plant must operate using one of the control decisions issued by the supervisor within the past cc time units. Importantly, this assumption is fundamental in many control system designs, as evidenced by the works, e.g., in [4], [7], [8], [16].

Based on Assumption A2, we adopt an over-approximation method to determine all potential control decisions that could impact the system's behavior. Specifically, under the control of a supervisor S, we consider an event to be viable to occur after a string if and only if it can occur under the control of at least one control decision issued by S within the past cc occurrences of the *tick* event, representing the passage of one time unit. This method considers all potential control decisions, accounting for those delayed in the control channel. Even when some decisions are lost during transmission, the effective decisions are still considered, allowing for accurate estimation of future behavior. As a result, delays and losses in the control channel are effectively handled. Subsequently, we define the language $\mathcal{L}(\Gamma_S/\mathfrak{N})$ generated by the compensated system $\Gamma_S/\mathfrak{N}$ (see Definition 5 on Page 7) , which emerges as an over-approximation of the actual behavior that can be generated by the closed-loop system $S/\mathfrak{N}$. We have summarized this discussion on the handling

of control delays and losses in Remark 1 on Page 7.

4.1) The authors mentioned that the advantages of this work compared to reference [27] lie in considering multiple observation channels and without setting FIFO in the control channel. Please explain the difficulties of considering multiple channels and provide an overview of solutions to these difficulties

Response: Thank you for this comment. Note that the original reference [27] in the text should now be identified as [23] in the revised paper. When considering multiple observation channels, several difficulties arise compared with a single-channel setup. First, each channel has its own unique characteristics, including an upper bound on observation delays and a potential for event losses. We need to track event transmission and delays in real time for each individual channel, taking into account its specific upper bound on delays and simulating any potential event losses within that channel. Additionally, event sequences may reach a supervisor in different orders from their actual order of occurrences in a plant.

To address these difficulties, we first define global observation channel configurations in Definition 3 on Page 3 of the paper to capture the dynamics of multiple observation channels. Then, we track event transmission and delays in real time for each individual channel by tracking global observation channel configurations through the operators *Add* (see Eq. (2) of the paper) and *Next* (Eq. (3)). Moreover, we use the operator *Deq* (Eq. (4)) to model any potential event losses within a channel. Finally, by considering all possible event outputs of multiple observation channels, we build an automaton G_e (see Definition 4 on Page 5) to capture changes in the order of observations by the supervisor.

4.2) In addition, please explain the advantages and disadvantages of without setting FIFO, and explain why FIFO setting is not required only in the control channel, while FIFO setting is still required in the observation channels.

Response: We appreciate your insights and have carefully considered your feedback. First, regarding the advantages of not employing FIFO in the control channel, this approach allows for increased system dynamism and responsiveness by processing and transmitting data in a non-sequential order. This flexibility enables more adaptive system behavior. However, it also presents challenges, as we need more complex algorithms to handle data processing without strict ordering. Additionally, systems relying on strict packet ordering, such as multimedia streaming, may not benefit from this approach due to the need for synchronization to avoid jitter or distortion.

Our decision not to implement FIFO in the control channel is rooted in the limitations inherent in the state estimation method proposed in [23], which is exclusively applicable to FIFO control channels. In contrast, the proposed approach aims for versatility, accommodating both FIFO and non-FIFO control channels. In Remark 1 on Page 7, we have discussed the fundamental principles behind adapting our approach to various control channel configurations. Furthermore, Appendix A.2 presents a simulation example that substantiates the versatility of the proposed method.

Regarding observation channels, FIFO setting is still required for several reasons. First, it is a mainstream approach widely adopted in the field, as evidenced by numerous studies that consider FIFO observation channels [4]–[8], [12], [14]–[16], [23], [24]. These studies demonstrate the effectiveness and widespread use of FIFO in observation channels, justifying our decision to focus on this approach. Second, incorporating non-FIFO observation channels would add further layers of complexity to our research, which already poses significant challenges. While we acknowledge the importance of researching non-FIFO observation channels, we have chosen to prioritize FIFO in this study due to its prevalence and page limit. Nevertheless, we intend to explore non-FIFO observation channels in future work.

5) Proof of Proposition 2 is too concise to understand. Provide the detailed proof process.

Response: We thank the reviewer for raising this point. We add a detailed proof as follows. Due to the page limit of the paper, all proofs are not presented in it. They can be accessed at `https://github.com/lahe1/SMC-proof/blob/main/SMC-proof.pdf`.

Proof for Proposition 2: ($\subseteq$) Consider any $\gamma \in \Gamma_S(\alpha)$. Due to Eq. (6), there necessarily exist observation strings $\alpha', \alpha'' \in \Sigma_{n,o}^*$ such that $\alpha = \alpha'\alpha''$, $\#_t(\alpha'') \leq cc$, and $\gamma = S(\alpha')$. According to Eq. (15), upon observing α', the decision γ paired with the maximum delay tolerance cc is sent to the configuration $\Lambda_S(\alpha')$, i.e., $(\gamma, cc) \in \Lambda_S(\alpha')$.

Now, as each *tick* event in α'' is observed, the delay tolerance decreases by one. Since there are $\#_t(\alpha'')$ such events and $\#_t(\alpha'') \leq cc$, it follows that after observing all of α'', the remaining delay tolerance is $cc - \#_t(\alpha'')$. Therefore, $(\gamma, cc - \#_t(\alpha'')) \in \Lambda_S(\alpha'\alpha'') = \Lambda_S(\alpha)$. By $cc - \#_t(\alpha'') \in \{0, 1, \ldots, cc\}$, we conclude

$$\gamma \in \{\gamma \in \Gamma \mid \exists l \in \{0, 1, \ldots, cc\} : (\gamma, l) \in \Lambda_S(\alpha)\}.$$

($\supseteq$) Conversely, suppose $\gamma \in \{\gamma \in \Gamma | \exists l \in \{0, 1, \ldots, cc\} : (\gamma, l) \in \Lambda_S(\alpha)\}$. Then, there exists some $l \in \{0, 1, \ldots, cc\}$ such that $(\gamma, l) \in \Lambda_S(\alpha)$. By the recursive definition of Λ_S, in the process of observing α, decision γ has necessarily been added to the configuration at some point, and its delay tolerance will be subsequently reduced to l. In particular, there exists a prefix α' of α such that $\gamma = S(\alpha')$ and the pair (γ, cc) is added to the configuration after observing α'. The remaining string α'' must contain exactly $cc - l$ events of *tick* in order to reduce the delay tolerance from cc to l. Therefore, one has $\#_t(\alpha'') = cc - l \leq cc$, satisfying the definition of augmented control decisions. Thus, we conclude $\gamma \in \Gamma_S(\alpha)$.

6) Some professional terms and symbols lack explanation such as "forcible events", and "symbol represented as a union symbol with a dot above it".

Response: We appreciate the reviewer's observation and would like to clarify the term "forcible events." This concept is extensively utilized in timed discrete event systems, as referenced in [9], [10], [11], [13], [16]. Essentially, a forcible event is a special event that preempts the tick of a global clock. This preemption is achieved by assigning a high priority to the event and implementing adaptable preemption strategies.

The purpose of this design is to enable the system to respond swiftly to urgent or vital control requirements, thereby enhancing its overall performance and reliability. It is important to note that although the forcible event exhibits preemptive behavior, we consider its preemption to be "weak" [9]. This means that its execution is not granted absolute priority but is determined based on the system's current states and demands. This flexibility allows the forcible event to be particularly effective in addressing complex control scenarios.

Furthermore, it is crucial to understand that preempting a tick does not equate to halting the system's clock. Rather, it signifies that an event's occurrence is compelled by timely preemption, overriding the usual tick cycle [13]. We next provide an example to illustrate the concept of forcible events.

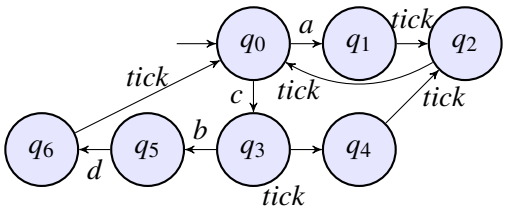

Figure 3: A TDES G: $\Sigma = \{a, b, c, d, tick\}$, $\Sigma_o = \{a, b, c, tick\}$, $\Sigma_{uo} = \{d\}$, $\Sigma_c = \{a, b\}$, $\Sigma_{uc} = \{c, d\}$, and $\Sigma_{for} = \{b\}$.

In the system as shown in Fig. 3, event b is considered a forcible event. As usual [11], a control decision takes the form $\gamma = (x, y)$, where $x \subseteq \Sigma \setminus \{tick\}$ represents the set of non-*tick* events enabled by γ, and $y \subseteq x \cap \Sigma_{for}$ denotes the set of events forced by γ, thereby preempting any pending *tick* event. Consider a state q_3 and a control decision $\gamma_1 = (\{b, c, d\}, \{b\})$. In this case, b is forced by γ_1 to occur, preempting the *tick* event. As a result, *tick* cannot occur at q_3. Now, consider another control decision $\gamma_2 = (\{b, c, d\}, \emptyset)$. Although γ_2 enables b to occur, b does not preempt *tick* since γ_2 does not force b.

Given that the forcible event is a common concept in the DES field, we were unable to provide a detailed explanation due to page limitations in the paper. However, in the third paragraph of the preliminaries section on Page 2, we briefly introduce forcible events as a means of preempting the tick event. For those unfamiliar with this concept, a comprehensive explanation can be found in the reference provided nearby.

Regarding the union symbol with a dot above it, it indicates the disjoint union of sets. This means that the sets being combined have no elements in common. For instance, when we write $\Sigma = \Sigma_{act} \dot{\cup} \{tick\}$, we are stating that Σ_{act} and $\{tick\}$ are disjoint sets, and their union comprises Σ. We explain this notation in the second paragraph on Page 2 of the paper.

7) In lines 43-44 on the right side of page 4, the formal expression of two sets is wrong. Please correct them and check the correctness of other formal expressions.

Response: Thank you for pointing out the potentially problematic sets. However, after careful consideration, we have decided to retain the current formulations of the two sets:

$$\Sigma_{so} = \{\sigma_o | \sigma \in \Sigma_{on}\}$$

$$\Sigma_{lo} = \{\sigma_l | \exists i \in I_o : \sigma \in \Sigma_{l,i}\}$$

We recognize that from a purely set-theoretic perspective, the expressions for these two sets might not seem elegant. However, within the DES community, they are widely accepted and commonly used. For instance, similar expressions can be found in Eq. (1) of [12], Definition 1 of [13], and Eq. (3) of [25]. Despite their apparent simplicity, these expressions effectively model a critical aspect: the supervisor's observations.

Specifically, when an event's transmission is completed within an observation channel, it is either observed by a supervisor or remains unobserved [12]. To handle these two outcomes, a common approach is to construct new events. Specifically, for any non *tick* observable event $\sigma \in \Sigma_{on}$, we derive two new events: σ_o and σ_l, by appending the subscript "o" and "l" to σ, respectively. Here, σ_o represents the successful observation of σ, while σ_l indicates the loss of observation of σ. Thus, $\Sigma_{so} = \{\sigma_o | \sigma \in \Sigma_{on}\}$ denotes the set of events representing successful observations. In addition, $\Sigma_{lo} = \{\sigma_l | \exists i \in I_o : \sigma \in \Sigma_{l,i}\}$ represents the set of events indicating loss of observations.

Due to the prevalence of the expressions for the above two sets in DES and their utility in modeling supervisor observations, we have retained them in our work. However, to enhance readability and comprehension, we have included explanations of successful observation events σ_o and lost observation events σ_l in the seventh paragraph on Page 4 of the paper. This additional context clarifies the meaning of the expressions of the sets for readers and facilitates a deeper understanding of the relevant concepts. Finally, we have carefully revisited all other formal expressions in the paper and made corrections wherever necessary.

8) Too many notations are introduced in this paper, a table providing a summary of the notations/acronyms is necessary to get a global view.

Response: We thank the reviewer for raising this point. We have provided a comprehensive Table I on Page 2 of the paper, summarizing all notations and acronyms used throughout the paper. We hope this table facilitates a global view and ease of understanding for the readers.

9) English needs more effort to improve the English grammar and sentence structure.

Response: Thank you for your comment. We have carefully revised the grammar and sentence structure throughout the text as requested.

We thank the reviewer for the thorough evaluation of our article and for providing valuable, high-quality comments that have been instrumental in improving the quality of our work. Thank you very much.

****

Reviewer's Comments to Authors:

Reviewer 2:

What are the contributions of the paper:

- *The notion of an NTDES is defined, where multipleobservation channels are considered.*
- *An observed automaton is developed to model the be-havior of the NTDES observed by a supervisor.*
- *An online over-approximation state estimation approachis proposed for the NTDES, which takes into account thecontrol decisions issued by a supervisor and the eventsobserved by the supervisor.*

Response: We appreciate the comments of the reviewer.

1) The motivation of this paper should be highlighted.

Response: We thank the reviewer for raising this point. The motivation behind this study stems from the compelling need for a versatile state estimation technique adaptable to both first-in-first-out (FIFO) and non-FIFO control channels while offering significantly improved precision compared with existing open-loop methods. We proceed to explain this further.

Within the research area of networked discrete event systems (NDESs), the development of state estimation techniques remains a key focus. Specifically, as highlighted in [25], online state estimation for determining the possible states of an NDES is crucial for verifying system properties such as opacity [27], observability [28], and detectability [29]. Furthermore, it plays a vital role in synthesizing a supervisor that enforces control specifications, as underscored in [8].

In developing state estimation methods for NDESs, accurately quantifying communication delays associated with data packets representing observable events or control decisions is crucial due to their significant impact on system performance [4]. A common approach involves counting the number of events that occur during the transmission of a packet through a communication channel [4]–[8], [19], [24]. However, this method relies on the assumption that event occurrences can be directly equated with time units, which may not hold true in practice as the intervals between event consecutive occurrences can be variable or irregular [9], [13].

To address this mismatch, an alternative idea for quantifying communication delays involves using the event *tick* from timed discrete event systems (TDESs), which can be directly associated with one time unit in practical applications [10]. This approach has motivated research efforts to extend NDESs to networked timed discrete event systems (NTDESs) by incorporating *tick* [14]–[17]. As a result, communication delays can be more accurately quantified, leading to the proposal of an open-loop state estimation method. However, this method does not consider the influence of control decisions. As a result, this method is inferior to those that consider the impact of control decisions, lacking the precision necessary for supervisor synthesis (see Appendix A.1 for more details).

Regarding the influence of control decisions, a state estimation method based on an over-approximation technique has been introduced in [8]. This method determines possible system

states by considering all delayed control decisions within the control channels. However, it uses event occurrences rather than *tick* to quantify communication delays, limiting its practical applicability [13], [23].

Recently, a state estimation method proposed in [23] has taken into account both the influence of control decisions and the event *tick*. However, its applicability is restricted to FIFO control channels, whereas non-FIFO channels are also widely used in practice [12]. Thus, the applicability of this method is restricted.

In light of the aforementioned limitations of existing state estimation methods, there is a compelling need for further refinement. Specifically, there is an urgent requirement for a versatile estimation technique that is applicable to both FIFO and non-FIFO control channels while offering significantly improved precision compared with existing open-loop methods. This need motivates us to design a state estimation method for NTDESs that incorporates the event *tick* and builds upon the over-approximation approach, aiming to meet the requirements of both control channel types. We have clarified this point in the paper (see the second through sixth paragraphs in the introductory section on Page 1 of the paper).

2) More details about the contributions should be provided by comparising with the current related works.

Response: Thank you for this comment. Our essential contribution is that the proposed over-approximation state estimation method strikes a balance between versatility and precision compared to the existing state estimation methods for NDESs.

We demonstrate the versatility of the proposed method. First, unlike the approach in [23], [24] limited to FIFO control channels, the proposed method, thanks to the over-approximation technique, can be applied to both FIFO and non-FIFO control channels. Second, the proposed method can effectively handle control delays and losses, by using the event *tick*. Hence, the proposed method is more practical and suitable for real-world systems than the time-agnostic state estimation methods in [4]–[8], [19], [24].

We demonstrate the precision of the proposed method. Considering the impact of control decisions on state estimation, the proposed method offers greater precision than the open-loop state estimation approaches proposed in [4]–[7], [14]–[16], [19].

Note that the proposed method is less precise compared with the method in [23], [24] due to its utilization of the over-approximation technique. Nevertheless, as mentioned earlier, the proposed method offers greater versatility. In summary, the proposed method strikes a balance between versatility and precision when compared with existing state estimation methods. We have clarified this point in the conclusion section on Page 12 of the paper. We also provide Table III on the same page, presenting a comprehensive overview of the proposed method's versatility and limitations in comparison to similar methods.

3) There are 12 examples in this paper, are they all necessary?

Response: We thank the reviewer for raising this point. Indeed, not all 12 examples are necessary. We carefully analyzed each one and decided to retain only five that we believe best represent the primary outcomes of our research. The other examples, while potentially interesting, were considered secondary and therefore not included in the final paper.

In particular, Example 1 on Page 4 of the paper demonstrates how we track global observation channel configurations using the operators *Add* (see Eq. (2)), *Next* (Eq. (3)), and *Deq* (Eq. (4)). Example 2 on Page 5 illustrates how we tackle observation delays and losses in modeling the open-loop behavior of an NTDES. Considering that the open-loop behavior constitutes a critical component of our state estimation method design, we deem these examples to be integral to this paper.

Regarding Example 3 on Page 7, its intention was to illustrate the impact of control delays and losses on the behavior of an NTDES, as well as to demonstrate the predictive effectiveness

of the proposed method under such conditions. We believe that this example not only enhances the comprehension of pertinent concepts but is also crucial for understanding the design of the proposed method. Therefore, we are of the opinion that it is worthwhile to retain this example in the revised version.

Examples 4 and 5 on Pages 10 and 11, respectively, specifically demonstrate the application of the proposed state estimation method for determining system states, significantly aiding readers' understanding. Therefore, it is meaningful to include them.

If the reviewer feels that some examples may not be as crucial as others, we would appreciate your specific feedback. We are open to making adjustments to improve the clarity and conciseness of this paper.

4) Simulation results are suggested to show the effectiveness of the developed methods.

Response: We thank the reviewer for raising this point. We have provided a simulation example accessible via `https://github.com/lahe1/SMC_simulation`. The technical specifications and results of the simulation are detailed in Appendix A.2 of this response letter.

Appendix A

A.1 An Introduction to Open-Loop State Estimation Methods

This section introduces the fundamental knowledge of open-loop state estimation methods for networked timed discrete event systems (NTDESs) proposed in [14]–[16].

Definition 1 *(Open-loop state estimation) Let $\mathfrak{N} = (G, oc, cc)$ be an NTDES, $G_e = (Q_e, \Sigma_e, \delta_e, q_{0_e})$ be the augmented plant for $\mathfrak{N}$, $\mathcal{L}(G_e)$ be the language generated by G_e, and $P_{e,o} : \Sigma_e^* \to \Sigma_{n,o}^*$ be the natural projection, where $\Sigma_{n,o}$ is the set of observable events of $\mathfrak{N}$. The open-loop state estimate of $\mathfrak{N}$ upon the occurrence of an observation string $\alpha \in P_{e,o}(\mathcal{L}(G_e))$ is defined by*

$$E_{ol}(\alpha) = \{q_e \in Q_e | \exists s_e \in \mathcal{L}(G_e) : P_{e,o}(s_e) = \alpha \wedge q_e = \delta_e(q_{0_e}, s_e)\}.$$

An open-loop state estimate represents a set of possible states for system G_e. The open-loop state estimation method infers the possible states of the system based on its model and available observation information without considering the impact of control decisions on system behavior. In this paper, the developed over-approximation state estimation method considers the influence of control decisions, which is reflected in the following definition (also see Definition 6 on Page 8 of the revised paper).

Definition 2 *Consider an NTDES $\mathfrak{N} = (G, oc, cc)$ and its augmented plant $G_e = (Q_e, \Sigma_e, \delta_e, q_{0_e})$. Let Γ_S be the augmented supervisor w.r.t. a supervisor S, as defined in Eq. (6). The state estimate of the compensated system $\Gamma_S/\mathfrak{N}$ upon the occurrence of an observation string $\alpha \in P_{e,o}(\mathcal{L}(\Gamma_S/\mathfrak{N}))$ is defined by*

$$E_{\Gamma_S}(\alpha) = \left\{ q_e \in Q_e \;\middle|\; \begin{array}{l} \exists s \in \mathcal{L}(\Gamma_S/\mathfrak{N}) : \\ P_{e,o}(s) = \alpha \wedge q_e = \delta_e(q_{0_e}, s) \end{array} \right\}.$$

Fact: The developed method is more precise than the open-loop state estimation method in the sense of $E_{\Gamma_S}(\alpha) \subseteq E_{ol}(\alpha)$, that is, $E_{\Gamma_S}(\alpha)$ contains fewer states that are unreachable by the system, as demonstrated in Fig. 4. Consequently, compared with supervisors designed using open-loop state estimation methods, the supervisor designed based on the proposed method enables the system to generate a larger language, that is, more permissive. This increased permissiveness enables the system to operate more efficiently and effectively under given control specifications [8]. This highlights the potential of the proposed approach in engineering applications of NTDESs.

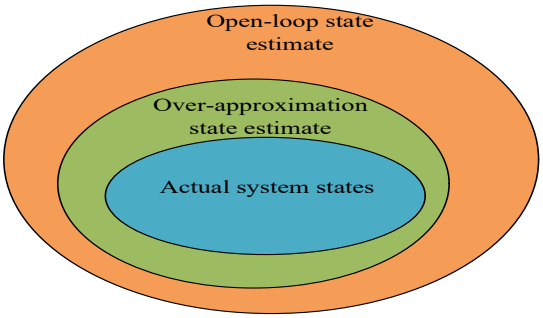

Figure 4: An illustration for the state estimation precision.

A.2 Proposed Method: Simulation and Analysis

This section introduces a simulation example that convincingly demonstrates the advantages of the proposed method. Accessible via `https://github.com/lahe1/SMC_simulation`, the simulation example highlights the proposed method's broader applicability when compared with the method presented in [23], [24], and showcases superior precision in contrast to the open-loop state estimation methods proposed in [14]–[16] (see Section A.1 in this Appendix for further details). Specifically, the simulation comprises five parts, as illustrated in Fig. 5. We proceed to introduce these parts as follows.

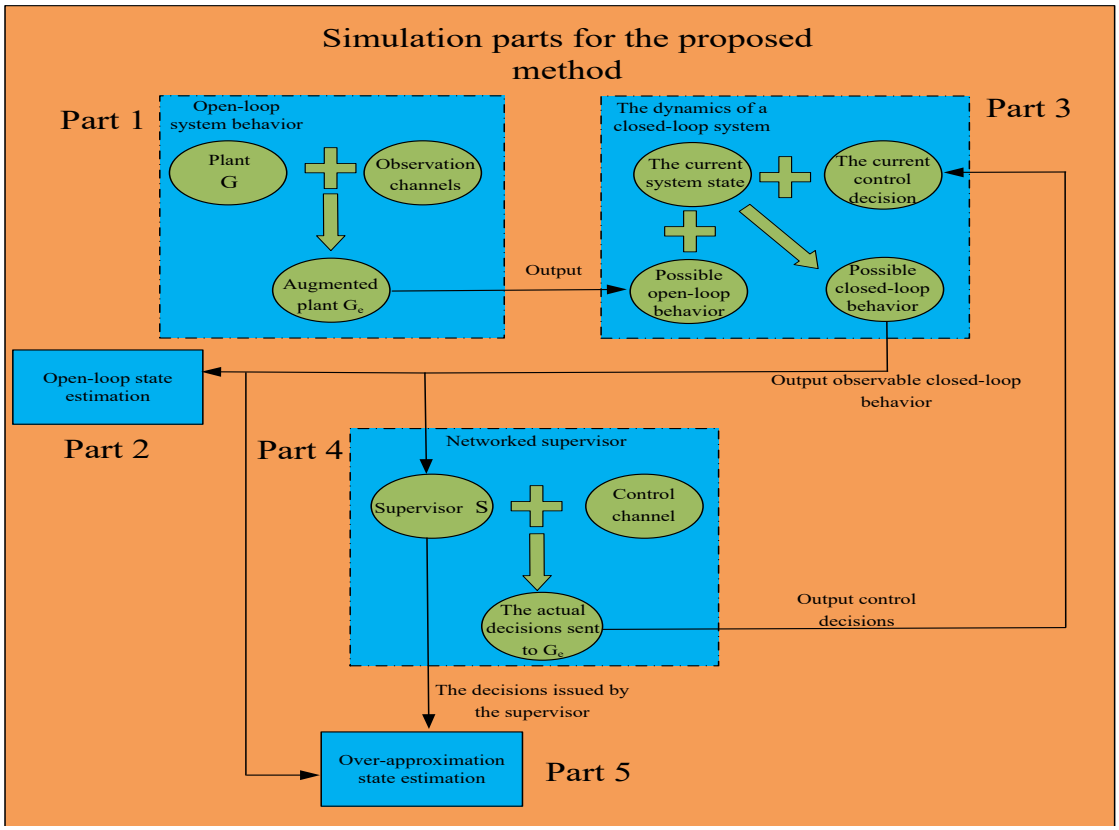

Figure 5: The simulation parts.

Part 1. Modeling open-loop system behavior:

- Given an NTDES $\Re = (G, oc, cc)$, this part augments plant G by incorporating the dynamics of the observation channels oc into an augmented plant G_e, which models the open-loop system behavior. The augmented plant is defined in Definition 4 on Page 5 of the paper.

Part 2. Computing open-loop state estimates:

- When an observable event occurs, this part computes the open-loop state estimate and then uses it to highlight the precision of the proposed approach.

Part 3. Modeling closed-loop system dynamics:

- This part models the dynamics of a closed-loop system $S/\Re$ controlled by a supervisor S. First, compute the active events of the current system state based on the augmented plant G_e, which represents the possible open-loop behavior at that state. Next, determine which of those active events can occur at the current state given the current control decision; these represent the possible closed-loop behavior at that state. Finally, from among those events, we randomly select one to occur.

Part 4. Modeling a networked supervisor:

- This part models a networked supervisor consisting of a supervisor S and a control channel with an upper bound cc on control delays and losses. Once a control decision is issued by S, it is first used to compute the current over-approximation state estimate. Subsequently, this decision is transmitted through the control channel. The control channel outputs control decisions to the augmented plant. The entire process simulates the dynamics of the networked supervisor.

- To highlight the broader applicability of the proposed method, we conduct simulations for a non-FIFO control channel (see A.2.1) and an FIFO one (see A.2.2).

Part 5. Implementing the over-approximation state estimation method:

- In this part, the developed method, as detailed in Algorithm 1 on Page 11 of the revised paper, is implemented to compute current over-approximation state estimates upon the occurrence of observable events and the issuance of control decisions.

For the simulation, we consider an NTDES $\Re = (G, oc, cc)$. The plant G is depicted in Fig. 6. The observation channels $oc = (oc_1, oc_2)$ satisfies $oc_1 = (\Sigma_{o,1} = \{a\}, \Sigma_{l,1} = \emptyset, N_{d,1} = 1)$ and $oc_2 = (\Sigma_{o,2} = \{b, c\}, \Sigma_{l,2} = \{c\}, N_{d,2} = 1)$. The value of bound cc is set to 0 and 1, separately. If he/she wishes to test larger values of cc, he/she can do so using the executable program named `SMC_simulation.exe`, accessed at `https://github.com/lahe1/SMC_simulation`. However, to reduce the number of pages in this response letter, we will not consider larger values of cc here.

The augmented plant $G_e = (Q_e, \Sigma_e, \delta_e, q_{0_e})$ for $\Re$ is shown in Fig. 7. Detailed information on the states of G_e is presented in Table 1, which tracks the states of G along with the corresponding global observation channel configurations. The set Γ of control decisions that a supervisor S can issue is displayed in Table 2. Note that the system state is initialized as q_{0e}, and the initial control decision used by G_e is randomly chosen from Γ to better demonstrate that the proposed method can deal with different supervisors.

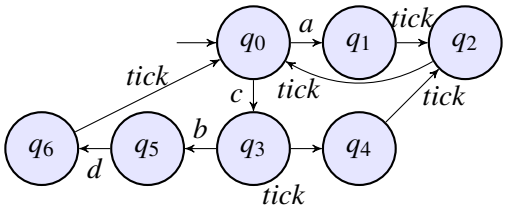

Figure 6: A TDES G: $\Sigma_o = \{a, b, c, tick\}$, $\Sigma_{uo} = \{d\}$, $\Sigma_c = \{a, b\}$, $\Sigma_{uc} = \{c, d\}$, and $\Sigma_{for} = \{b\}$.

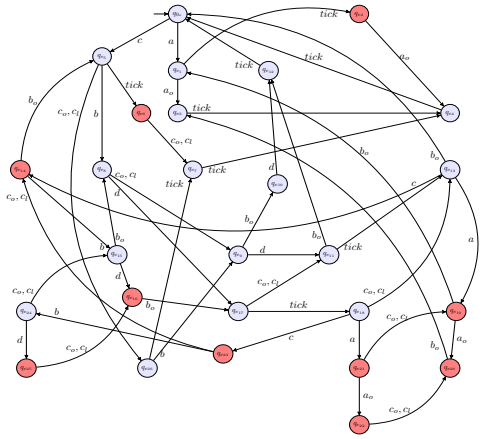

Figure 7: The augmented plant G_e for the NTDES.

Table 1: States of the augmented plant G_e.

$q_{0_e} = (q_0, (\varepsilon, \varepsilon))$	$q_{e_1} = (q_1, ((a, 1), \varepsilon))$
$q_{e_2} = (q_2, ((a, 0), \varepsilon))$	$q_{e_3} = (q_1, (\varepsilon, \varepsilon))$
$q_{e_4} = (q_2, (\varepsilon, \varepsilon))$	$q_{e_5} = (q_3, (\varepsilon, (c, 1)))$
$q_{e_6} = (q_4, (\varepsilon, (c, 0)))$	$q_{e_7} = (q_4, (\varepsilon, \varepsilon))$
$q_{e_8} = (q_5, (\varepsilon, (c, 1)(b, 1)))$	$q_{e_9} = (q_5, (\varepsilon, (b, 1)))$
$q_{e_{10}} = (q_5, (\varepsilon, \varepsilon))$	$q_{e_{11}} = (q_6, (\varepsilon, (b, 1)))$
$q_{e_{12}} = (q_6, (\varepsilon, \varepsilon))$	$q_{e_{13}} = (q_0, (\varepsilon, (b, 0)))$
$q_{e_{14}} = (q_3, (\varepsilon, (b, 0)(c, 1)))$	$q_{e_{15}} = (q_5, (\varepsilon, (b, 0)(c, 1)(b, 1)))$
$q_{e_{16}} = (q_6, (\varepsilon, (b, 0)(c, 1)(b, 1)))$	$q_{e_{17}} = (q_6, (\varepsilon, (c, 1)(b, 1)))$
$q_{e_{18}} = (q_0, (\varepsilon, (c, 0)(b, 0)))$	$q_{e_{19}} = (q_1, ((a, 1), (b, 0)))$
$q_{e_{20}} = (q_1, (\varepsilon, (b, 0)))$	$q_{e_{21}} = (q_1, ((a, 1), (c, 0)(b, 0))$
$q_{e_{22}} = (q_1, (\varepsilon, (c, 0)(b, 0)))$	$q_{e_{23}} = (q_3, (\varepsilon, (c, 0)(b, 0)(c, 1)))$
$q_{e_{24}} = (q_5, (\varepsilon, (c, 0)(b, 0)(c, 1)(b, 1)))$	$q_{e_{25}} = (q_6, (\varepsilon, (c, 0)(b, 0)(c, 1)(b, 1)))$
$q_{e_{26}} = (q_3, (\varepsilon, \varepsilon))$	

Table 2: Control decision set Γ.

$\gamma_1 = (\{c, d, a_o, b_o, c_o, c_l\}, \emptyset)$
$\gamma_2 = (\{c, d, a_o, b_o, c_o, c_l, a\}, \emptyset)$
$\gamma_3 = (\{c, d, a_o, b_o, c_o, c_l, b\}, \emptyset)$
$\gamma_4 = (\{c, d, a_o, b_o, c_o, c_l, b\}, \{b\})$
$\gamma_5 = (\{c, d, a_o, b_o, c_o, c_l, a, b\}, \emptyset)$
$\gamma_6 = (\{c, d, a_o, b_o, c_o, c_l, a, b\}, \{b\})$

A.2.1 Proposed Method: Simulation and Analysis for non-FIFO Control Channels

The C++ code for the simulation can be accessed at `https://github.com/lahe1/SMC_simulation/blob/master/SMC_simulation/SMC_simulation.cpp`, and the code's structure specifically for the non-FIFO control channel is illustrated in the flowchart shown in Fig. 8.

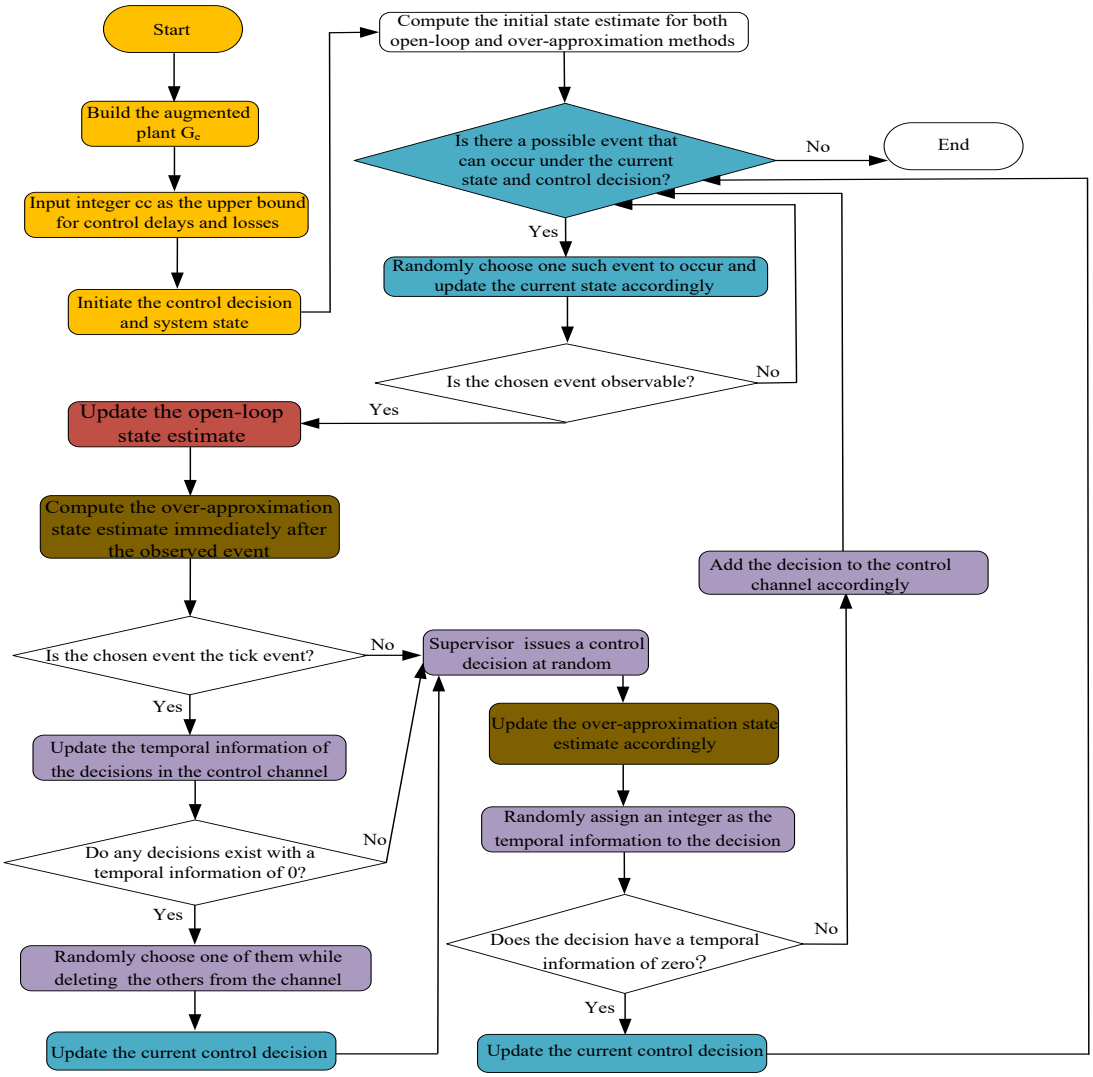

Figure 8: The algorithm flowchart for the simulation with non-FIFO control channel.

First, we construct the augmented plant G_e and set the initial system state to q_{0e}. Then, we input a control and delay bound cc. These steps are detailed in lines 1549–1938 of the provided C++ code. Next, we randomly select an initial control decision, implemented in line 718.

Subsequently, we compute the initial open-loop state estimate and the initial over-approximation state estimate, associated with lines 691 and 734, respectively. Then, we execute line 774 to ascertain whether closed-loop behavior can occur given the current state and control decision. If such behavior cannot occur, we stop the simulation. Otherwise, we randomly select a feasible event to occur using line 796 and update the current state accordingly using line 804. Determining whether the selected event is observable is implemented in line 829.

After selecting an observable event, we accordingly compute the open-loop state estimate (line

850), followed by the computation of the over-approximation state estimate (line 855). The latter computation is based on the operator O_r as defined in Eqs. (11)–(13) of the revised paper.

(1) If the selected event is not *tick* (as determined in line 885), we randomly select a control decision from the control decision set Γ to model the supervisor's issuing behavior (line 1006). Then, we incorporate the newly issued control decision to update the over-approximation state estimate (line 1017).

To model control delays, we assign a random integer value between 0 and *cc* (inclusive) as the temporal information to any control decision issued (line 1009). This temporal information indicates the number of *tick* events that must occur before the completion of the corresponding decision's transmission. If the assigned value for a control decision is zero, it indicates the absence of control delay and loss. As a result, the transmission of that decision through the channel occurs instantaneously, and we can immediately apply this decision to update the current control decision. After that, we ascertain whether closed-loop behavior can occur based on the current state and control decision. The implementation of the decision handling without delays and losses can be found in lines 1020–1026. On the other hand, if the assigned value is greater than zero, indicating a control delay, the decision is sent to the control channel. The current control decision remains unchanged. Here, again, we check for the possibility of closed-loop behavior based on the current state and control decision (lines 1027–1036). The entire process for modeling control delays is illustrated in Fig. 9.

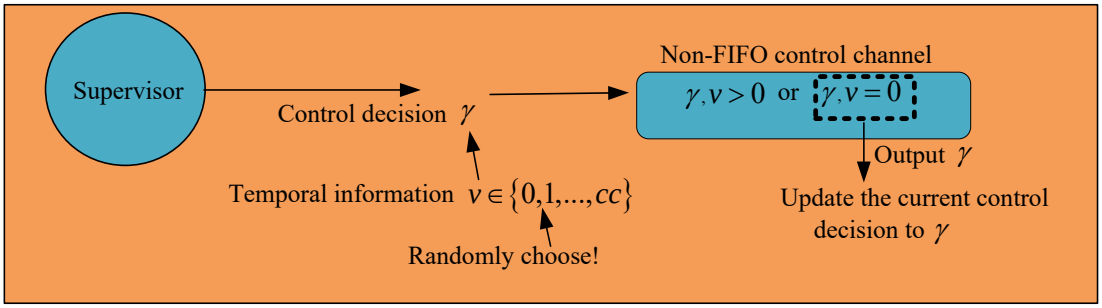

Figure 9: An illustration of modeling control delays.

(2) If the selected event is *tick*, we will update the temporal information for each control decision in the control channel by decreasing it by one (lines 885–893). After that, the control channel outputs all control decisions with temporal information valued at zero. To simulate control losses bounded by *cc*, we randomly select only one control decision from the output and use it to update the current control decision while discarding the others (lines 907–976). Subsequently, we randomly select a control decision to model the supervisor's issuing behavior after the occurrence of *tick*, and then execute the same steps as mentioned earlier. The entire process for modeling control losses is illustrated in Fig. 10.

Since the temporal information values we assign to control decisions are randomly selected from the range of 0 to *cc*, the decisions in the control channel are not required to follow an FIFO order when being output. Therefore, we have successfully simulated the delays and losses in a non-FIFO control channel, as desired.

Now, we present the simulation result. For brevity and to conserve the number of pages of the response letter, the tables below present data from the first ten observable events for each *cc* value tested. It should be noted that, as mentioned earlier, we use a random approach to simulate system dynamics, control decision issuance, and control delays and losses. Hence, even for the same *cc*, the data obtained from the simulation do not need to be the same.

We first provide simulation data with $cc = 0$, indicating no control delays and losses, as

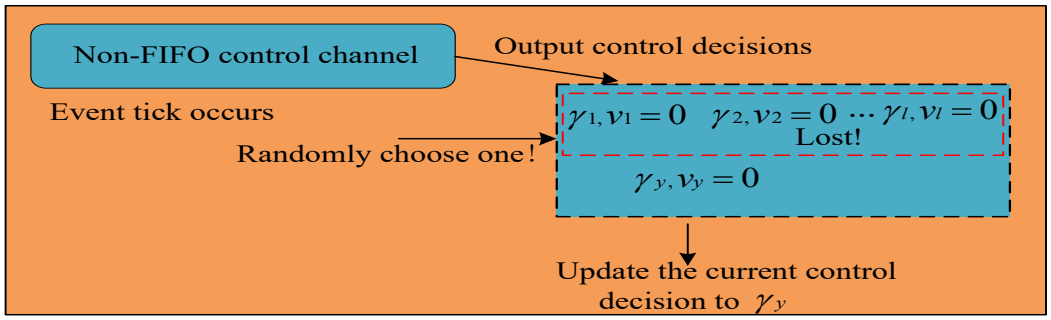

Figure 10: An illustration of modeling control losses.

illustrated in Table 3. Table 4 is presented to clarify the meanings of each entry listed in Table 3. For instance, let us sequentially introduce the entries of the forth row in Table 3. (1) *tick* represents the event observed by a supervisor. (2) γ_1 is the control decision issued by the supervisor after observing *tick*. (3) q_{e4} represents a possible system's state after the occurrence of *tick*, estimating under the open-loop state estimation method. (4) γ_1 denotes the control decision in use by the plant G_e. (5) "No" signifies the absence of control losses after the occurrence of *tick*. (6) q_{e4} (in the sixth column) represents a possible system's state after the occurrence of *tick* and the issuance of the decision γ_1, estimating under the over-approximation state estimation method. (7) q_{e4} is the actual state in which the system has reached until the next observable event.

Table 3: Simulation result: Non-FIFO and $cc = 0$.

$\Sigma_{n,o}$	S	E_{ol}	Γ	Losses	E_{Γ_S}	G_e
ε	γ_1	q_{0e},q_{e1},q_{e5},q_{e8}, q_{e26},q_{e9},q_{e17},q_{e11}	γ_1	No	q_{0e}, q_{e5}, q_{e26}	q_{0e}, q_{e5}
tick	γ_3	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_3	No	q_{e6},q_{e7}	q_{e6},q_{e7}
tick	γ_1	q_{e4}	γ_1	No	q_{e4}	q_{e4}
tick	γ_4	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_4	No	q_{0e}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e}, q_{e5}, q_{e26}, q_{e9}
b_o	γ_4	q_{e10},q_{e12}	γ_4	No	q_{e10},q_{e12}	q_{e10},q_{e12}
tick	γ_2	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_2	No	q_{0e}, q_{e1}, q_{e5}, q_{e26}	q_{0e},q_{e5}
tick	γ_5	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_5	No	q_{e2}, q_{e6}, q_{e7}	q_{e6}
c_o	γ_4	q_{e7}, q_{e13}, q_{e19}, q_{e14}, q_{e15}, q_{e16}	γ_4	No	q_{e7}	q_{e7}
tick	γ_1	q_{e4}	γ_1	No	q_{e4}	q_{e4}
tick	γ_4	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_4	No	q_{0e}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e}, q_{e5}, q_{e8},q_{e17}
tick	γ_1	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_1	No	q_{e18}, q_{e13}, q_{e23}, q_{e14}	q_{e18}
c_o	γ_2	q_{e7}, q_{e13}, q_{e19}, q_{e14}, q_{e15}, q_{e16}	γ_2	No	q_{e13}, q_{e14},q_{e19}	q_{e13}

Notice that every entry in the second column of Table 3 is identical to the corresponding entry in the fourth column of the same row. Additionally, all entries in the fifth column are 'no.' The two observations indicate that the simulation has successfully implemented the scenario where $cc = 0$, i.e., the control channel has no delays or losses.

According to Table 3, the states that the system has actually reached are encompassed by both the open-loop and the over-approximation state estimates. Therefore, both methods can estimate the plant states. However, the proposed method is significantly more precise than the open-loop

Table 4: Explanation of Table 3.

Column $\Sigma_{n,o}$: The events observed by the supervisor
Column S: The control decisions issued by the supervisor
Column E_{ol}: The open-loop state estimates after observing the events listed in the same rows
Column Γ: The control decisions currently in use
Column Losses: The control decisions currently lost
Column E_{Γ_S}: The most recently over-approximation state estimates
Column G_e: The actual states for the system

method. For example, in the third row, the proposed method accurately estimates the actual system states, whereas the open-loop method estimates a large number of states that the system will not actually reach. We next provide simulation data with $cc = 1$, indicating that the control channel has control delays and losses bounded by 1, as illustrated in Table 5.

Table 5: Simulation result: Non-FIFO and $cc = 1$.

$\Sigma_{n,o}$	S	E_{ol}	Γ	Losses	E_{Γ_S}	G_e
ε	γ_1	$q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}$	γ_1	No	$q_{0e}, q_{e5}, q_{e26},$	q_{0e}, q_{e5}
$tick$	γ_2	$q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}$	γ_1	No	q_{e6}, q_{e7}	q_{e6}
c_o	γ_2	$q_{e7}, q_{e13}, q_{e19}, q_{e14}, q_{e15}, q_{e16}$	γ_1	No	q_{e7}	q_{e7}
$tick$	γ_1	q_{e4}	γ_1	γ_2	q_{e7}	q_{e7}
$tick$	γ_4	$q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}$	γ_1	No	q_{0e},q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e},q_{e5}
c_o	γ_3	q_{e26}, q_{e9}, q_{e11}	γ_1	No	q_{e26}, q_{e9}, q_{e11}	q_{e26}
$tick$	γ_4	$q_{e7}, q_{e13}, q_{e14}, q_{e19}, q_{e15}, q_{e16}$	γ_4	No	$q_{e7}, q_{e13}, q_{e14}, q_{e15}, q_{e16}$	q_{e7}
$tick$	γ_4	q_{e4}	γ_4	No	q_{e4}	q_{e4}
$tick$	γ_1	$q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}$	γ_4	No	q_{0e}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e}, q_{e5}, q_{e26}, q_{e9}
b_o	γ_5	q_{e10}, q_{e12}	γ_5	No	q_{e10}, q_{e12}	q_{e10}, q_{e12}
$tick$	γ_1	$q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}$	γ_1	No	q_{0e}, q_{e1}, q_{e5}, q_{e8}, $q_{e26}, q_{e9}, q_{e17}, q_{e11}$	q_{0e},q_{e5}
c_o	γ_2	q_{e26}, q_{e9}, q_{e11}	γ_1	No	q_{e26}, q_{e9}, q_{e11}	q_{e26}

According to Table 5, the simulation has successfully modeled control delays and losses bounded by $cc = 1$ in the control channel. Moreover, the non-FIFO property of the control channel has also been well simulated. For instance, let us observe rows 10 and 11. Control decision γ_5 is received by the plant G_e first, despite γ_1 entering the control channel earlier.

By setting cc to 0 and 1, the simulation confirms that the proposed method can be effectively applied to the non-FIFO control channel for estimating system states. It also shows that the proposed method is more precise than the open-loop approach.

A.2.2 Proposed Method: Simulation and Analysis for FIFO Control Channels

The simulation code's structure specifically for the FIFO control channel is illustrated in the flowchart shown in Fig. 11. Specifically, the steps enclosed within the red dashed box are used to simulate the properties of an FIFO control channel and differ from those in Fig. 8, which are for simulating a non-FIFO channel. All other steps are the same.

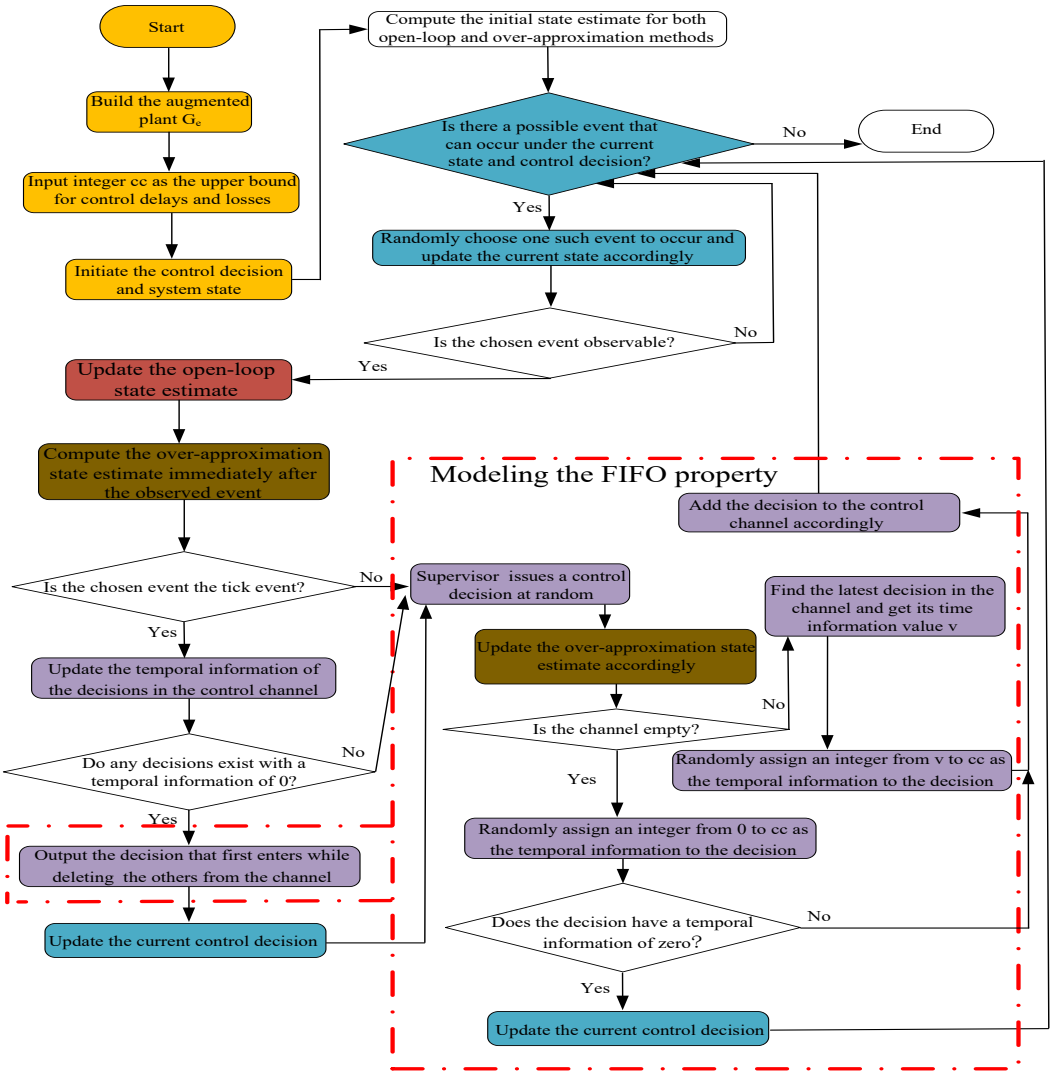

Figure 11: The algorithm flowchart for the simulation with FIFO control channel.

Let us consider the steps outlined within the red dashed box. When a supervisor issues a control decision γ, to model control delays, we initially check whether the control channel is empty (implemented in line 1440 of the C++ code available at `https://github.com/lahe1/SMC_simulation/blob/master/SMC_simulation/SMC_simulation.cpp`).

(1) If the control channel is empty, we assign a randomly selected value of temporal information to the issued decision γ in the range from 0 to cc (line 1442).

(2) If the channel is not empty, we identify the last decision γ' entered and retrieve its temporal information value v. Subsequently, as implemented in line 1467, we assign a randomly selected value to the issued decision γ within the range of v to cc. This approach ensures that the temporal information value assigned to γ is greater than or equal to v, thus simulating the FIFO characteristic where γ completes its transmission in the channel later than γ'.

In summary, the method for modeling control delays when the channel is empty is the same as illustrated in Fig. 9, while the case that the channel is not empty is shown in Fig. 12.

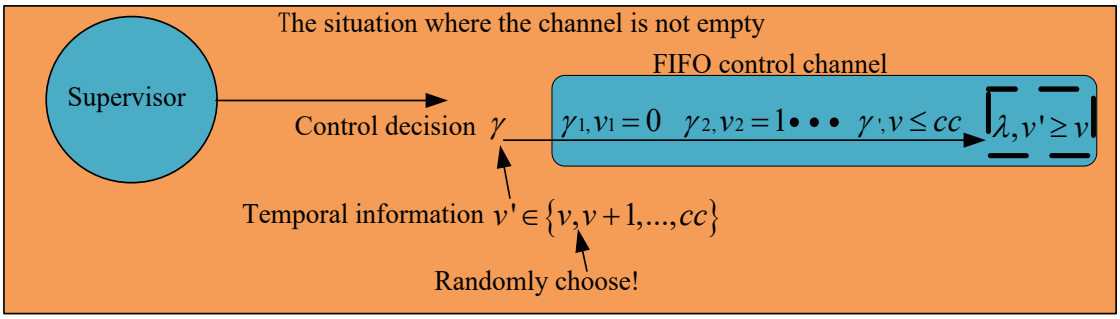

Figure 12: An illustration of modeling control delays for FIFO channel.

When the event *tick* occurs, we will update the temporal information for each control decision in the control channel (lines 1308–1324). After that, the control channel outputs all control decisions with temporal information valued at zero (line 1334). To simulate control losses, we select the control decision that first enters the channel from the output and use it to update the current control decision, while discarding the rest (lines 1353 and 1388). The entire method for modeling control losses is illustrated in Fig. 13. We finally provide simulation data with $cc = 1$, as illustrated in Table 6

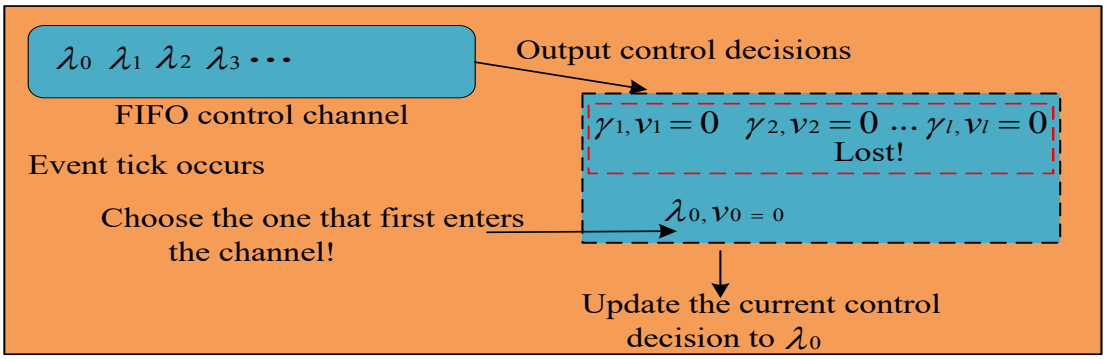

Figure 13: An illustration of modeling control losses for FIFO channel.

According to Table 6, the simulation has successfully modeled control delays and losses bounded by $cc = 1$ in the control channel. The FIFO characteristic of the control channel has also been well modeled, as no cases were found in which a later control decision completed transmission before an earlier one. Moreover, the simulation confirms that the proposed method can be effectively applied to the FIFO control channel for estimating system states. It also shows that the proposed method is more precise than the open-loop approach.

Table 6: Simulation result: FIFO and $cc = 1$.

$\Sigma_{n,o}$	S	E_{ol}	Γ	Losses	E_{Γ_S}	G_e
ε	γ_1	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_1	No	q_{0e}, q_{e5},q_{e26}	q_{0e}, q_{e5}
tick	γ_4	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_4	No	q_{e6},q_{e7}	q_{e6}
c_o	γ_2	q_{e7}, q_{e13}, q_{e19}, q_{e14}, q_{e15}, q_{e16}	γ_2	No	q_{e7}	q_{e7}
tick	γ_6	q_{e4}	γ_2	No	q_{e4}	q_{e4}
tick	γ_6	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_6	No	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e}, q_{e5}, q_{e8}, q_{e17}, q_{e11}
tick	γ_2	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_6	No	q_{e2}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	q_{e13},q_{e14}
b_o	γ_3	q_{0e}, q_{e5}, q_{e1}, q_{e8}, q_{e17}, q_{e26}, q_{e9}, q_{e11}	γ_6	No	q_{0e}, q_{e5}, q_{e1}, q_{e8}, q_{e17}, q_{e26}, q_{e9}, q_{e11}	q_{e5}
c_o	γ_5	q_{e26}, q_{e9}, q_{e11}	γ_6	No	q_{e26}, q_{e9}, q_{e11}	q_{e26}, q_{e9}
b_o	γ_6	q_{0e},q_{e12}	γ_6	No	q_{0e},q_{e12}	q_{0e},q_{e12}
tick	γ_3	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	γ_2	γ_3,γ_5,γ_6	q_{0e}, q_{e1}, q_{e5}, q_{e8}, q_{e26}, q_{e9}, q_{e17}, q_{e11}	q_{0e}, q_{e1}, q_{e26}
tick	γ_2	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	γ_2	No	q_{e2}, q_{e6}, q_{e7}, q_{e18}, q_{e13}, q_{e21}, q_{e23}, q_{e14}, q_{e19}, q_{e24}, q_{e15}, q_{e25}, q_{e16}	q_{e7}
tick	γ_1	q_{e4}	γ_1	No	q_{e4}	q_{e4}

A.2.3 Simulation Conclusion

Through the simulation, we have verified the applicability of the proposed over-approximation state estimation method to both non-FIFO (see A.2.1) and FIFO (see A.2.2) control channels. This demonstrates its broader potential compared with the approach presented in [23], [24], which is limited to FIFO scenarios.

Moreover, the simulation shows that the proposed method, considering the impact of control decisions on state estimation, offers greater precision than the open-loop state estimation approach proposed in [4]–[7], [14]–[16], [19].

Additionally, it is shown that the proposed method can effectively handle control delays and losses, taking into account the passage of time modeled by the occurrences of the event *tick*. This is crucial since real-world delays and losses are typically measured by time passage [9], [10]. Therefore, the proposed method is more practical and suitable for real-world systems than the time-agnostic state estimation methods in [4]–[8], [19], [24]. To provide a concise overview and aid in the comprehension of the suitability of the various methods, we have compiled Table 7.

Table 7: Overview of the existing state estimation methods.

Citation	Timed	FIFO	non-FIFO	Control
[4],[5],[6],[7]	✗	✓	✓	✗
[8]	✗	✓	✓	✓
[14], [15],[16]	✓	✓	✓	✗
[19]	✗	✓	✓	✗
[23]	✓	✓	✗	✓
[24]	✗	✓	✗	✓
The proposed method	✓	✓	✓	✓

References

[1] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems,* 2nd ed. Springer, 2008.

[2] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Cont. Opt.,* vol. 25, no. 1, pp. 206–230, 1987.

[3] X. Yin and S. Lafortune, "Synthesis of maximally permissive supervisors for partially-observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1239–1254, 2016.

[4] F. Lin, "Control of networked discrete event systems: Dealing with communication delays and losses," *SIAM J. Cont. Opt.,* vol. 52, no. 2, pp. 1276–1298, 2014.

[5] S. Shu and F. Lin, "Supervisor synthesis for networked discrete event systems with communication delays," *IEEE Trans. Autom. Control,* vol. 60, no. 8, pp. 2183–2188, 2015.

[6] S. Shu and F. Lin, "Deterministic networked control of discrete event systems with nondeterministic communication delays," *IEEE Trans. Autom. Control,* vol. 62, no. 1, pp. 190–205, 2017.

[7] S. Shu and F. Lin, "Predictive networked control of discrete event systems," *IEEE Trans. Autom. Control,* vol. 62, no. 9, pp. 4698–4705, 2017.

[8] Z. Liu, X. Yin, S. Shu, F. Lin, and S. Li, "Online supervisory control of networked discrete-event systems with control delays," *IEEE Trans. Autom. Control,* vol. 67, no. 5, pp. 2314–2329, 2022.

[9] B. A. Brandin and W. M. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Trans. Autom. Control,* vol. 39, no. 2, pp. 329–342, 1994.

[10] F. Lin and W. M. Wonham, "Supervisory control of timed discrete-event systems under partial observation," *IEEE Trans. Autom. Control,* vol. 40, no. 3, pp. 558–562, 1995.

[11] S. Takai and T. Ushio, "A new class of supervisors for timed discrete event systems under partial observation," *Discrete Event Dyn. Syst.,* vol. 16, no. 2, pp. 257–278, 2006.

[12] M. V. S. Alves, L. K. Carvalho, and J. C. Basilio, "Supervisory control of networked discrete event systems with timing structure," *IEEE Trans. Autom. Control,* vol. 66, no. 5, pp. 2206–2218, 2021.

[13] A. Rashidinejad, M. Reniers, and L. Feng, "Supervisory control of timed discrete-event systems subject to communication delays and non-fifo observations," in *14th International Workshop on Discrete Event Systems*, pp. 456–463, 2018.

[14] C. Miao, S. Shu, and F. Lin, "State estimation for timed discrete event systems with communication delays," in *Proc. Chinese Automation Congress*. IEEE, 2017, pp. 2721–2726.

[15] C. Miao, S. Shu, and F. Lin, "Predictive supervisory control for timed discrete event systems under communication delays," in *Proc. IEEE 58th Conference on Decision and Control*, 2019, pp. 6724–6729.

[16] B. Zhao, F. Lin, C. Wang, X. Zhang, M. P. Polis, and L. Y. Wang, "Supervisory control of networked timed discrete event systems and its applications to power distribution networks," *IEEE Trans. Contr. Netw. Syst.,* vol. 4, no. 2, pp. 146–158, 2017.

[17] S. J. Park and K. H. Cho, "Nonblocking supervisory control of timed discrete event systems under communication delays: The existence conditions," *Automatica*, vol. 44, no. 4, pp. 1011–1019, 2008.

[18] R. Tai, L. Lin, Y. Zhu, and R. Su, "A new modeling framework for networked discrete-event systems," *Automatica*, vol. 138, 2022.

[19] Y. Yao, Y. Tong, and H. Lan, "Initial-state estimation of multi-channel networked discrete event systems," *IEEE Control Syst. Lett*., vol. 4, no. 4, pp. 1024–1029, 2020.

[20] C. Gu, Z. Y. Ma, Z. W. Li, and A. Giua, "Verification of nonblockingness in bounded Petri nets with min-max basis reachability graphs," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 52, no. 10, pp. 6162–6173, 2022.

[21] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "Critical observability of discrete-event systems in a Petri net framework," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 52, no. 5, pp. 2789–2799, 2022.

[22] Y. F. Chen, Y. T. Li, Z. W. Li, and N. Q. Wu, "On optimal supervisor design for discrete-event systems modeled with Petri nets via constraint simplification," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 52, no. 6, pp. 3404–3418, 2022.

[23] Y. F. Hou, Y. F. Ji, G. Wang, C. Y. Weng, and Q. D. Li, "Modeling and state estimation for supervisory control of networked timed discrete-event systems and their application in supervisor synthesis," *International Journal of Control*, 2023, doi: 10.1080/00207179.2023.2204382.

[24] Y. F. Hou, Y. F. Ji, G. Wang, C. Y. Weng, and Q. D. Li, "Online state estimation for supervisor synthesis in discrete-event systems with communication delays and losses," *IEEE Trans. Contr. Netw. Syst.,* pp. 1–12, 2023, doi: 10.1109/TCNS.2023.3280461.

[25] M. V. S. Alves and J. C. Basilio, "State estimation and detectability of networked discrete event systems with multi-channel communication networks," *IEEE Trans. Autom. Sci. Eng.,* pp. 1–16, 2023, doi: 10.1109/TASE.2023.3265846.

[26] Z. Y. Xiang, Y. F. Chen, N. Q. Wu, and Z. W. Li, "Supplement material for the paper." [Online]. Available: https://github.com/lahe1/SMC-proof/blob/main/SMC-proof.pdf

[27] S. Lafortune, F. Lin, and C. N. Hadjicostis, "On the history of diagnosability and opacity in discrete event systems," *Annu. Rev. Control,* vol. 45, pp. 257–266, 2018.

[28] W. M. Wonham and K. Cai, *Supervisory Control of Discrete-Event Systems*. Heidelberg, Germany: Springer, 2019.

[29] T. Masopust and X. Yin, "Complexity of detectability, opacity and A-diagnosability for modular discrete event systems," *Automatica*, vol. 101, pp. 290–295, 2019.

[30] Z. Liu, J. Hou, X. Yin, and S. Li, "Modeling and analysis of networked supervisory control systems with multiple control channels," in *Proc. IEEE 60th Conference on Decision and Control*, 2021, pp. 316–323.