Simulation Example

1 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 detailed in [14]–[16]. Specifically, the simulation comprises five parts, as illustrated in Fig. 1. We proceed to introduce these parts as follows.

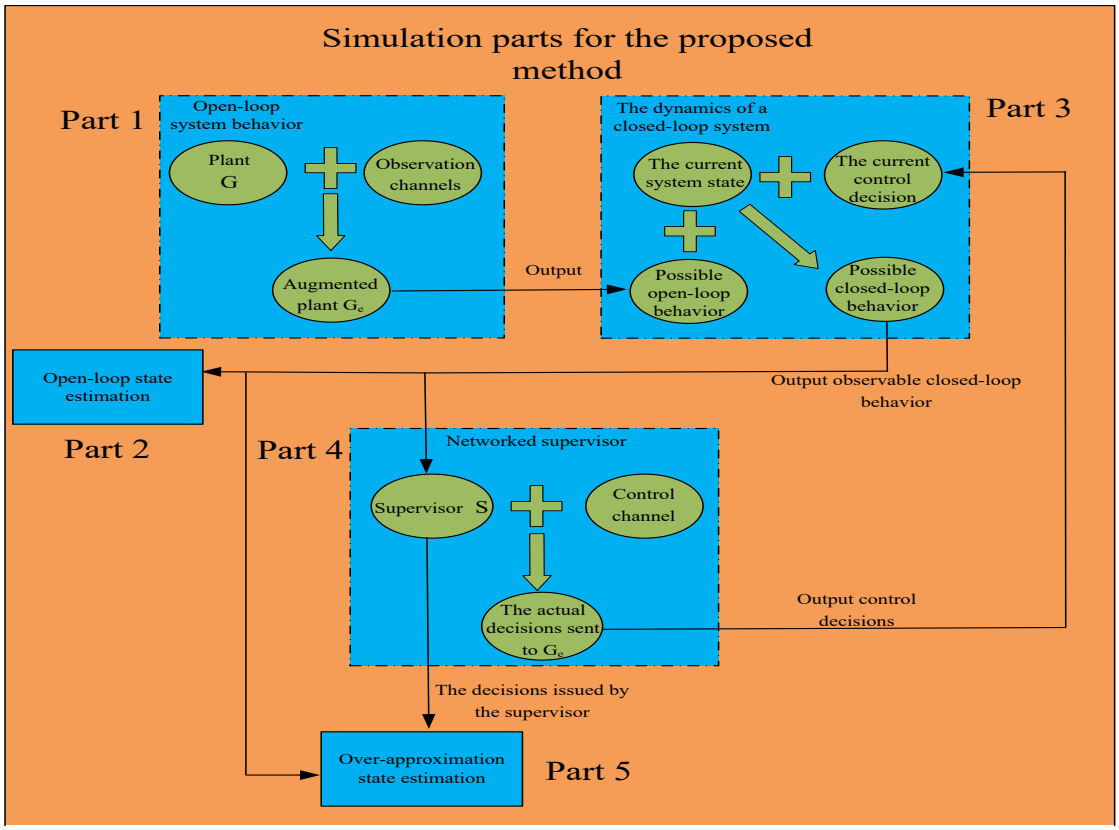

Figure 1: The simulation parts.

Part 1. Modeling open-loop system behavior:

- Given an NTDES $\Omega = (G, oc, cc)$, 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/Ω 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 1.1) and an FIFO one (see 1.2).

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 $\Omega = (G, oc, cc)$. The plant G is depicted in Fig. 2. 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 Ω is shown in Fig. 3. 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.

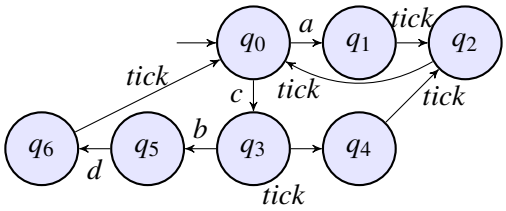

Figure 2: A TDES G: $\Sigma_o = \{a, b, c, tick\}$, $\Sigma_{uo} = \{d\}$, $\Sigma_c = \{a, b\}$, $\Sigma_{uc} = \{c, d\}$, and $\Sigma_{for} = \{b\}$.

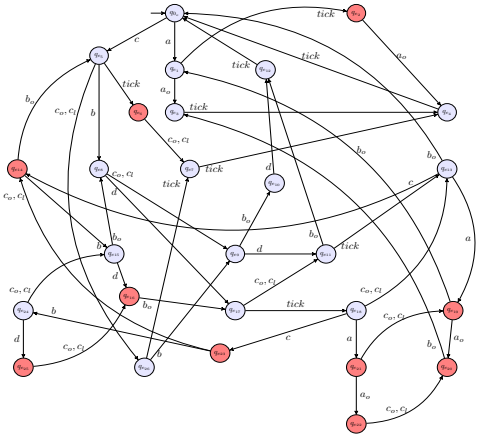

Figure 3: The augmented plant G_e for the NTDES.

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.

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

1.1 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. 4.

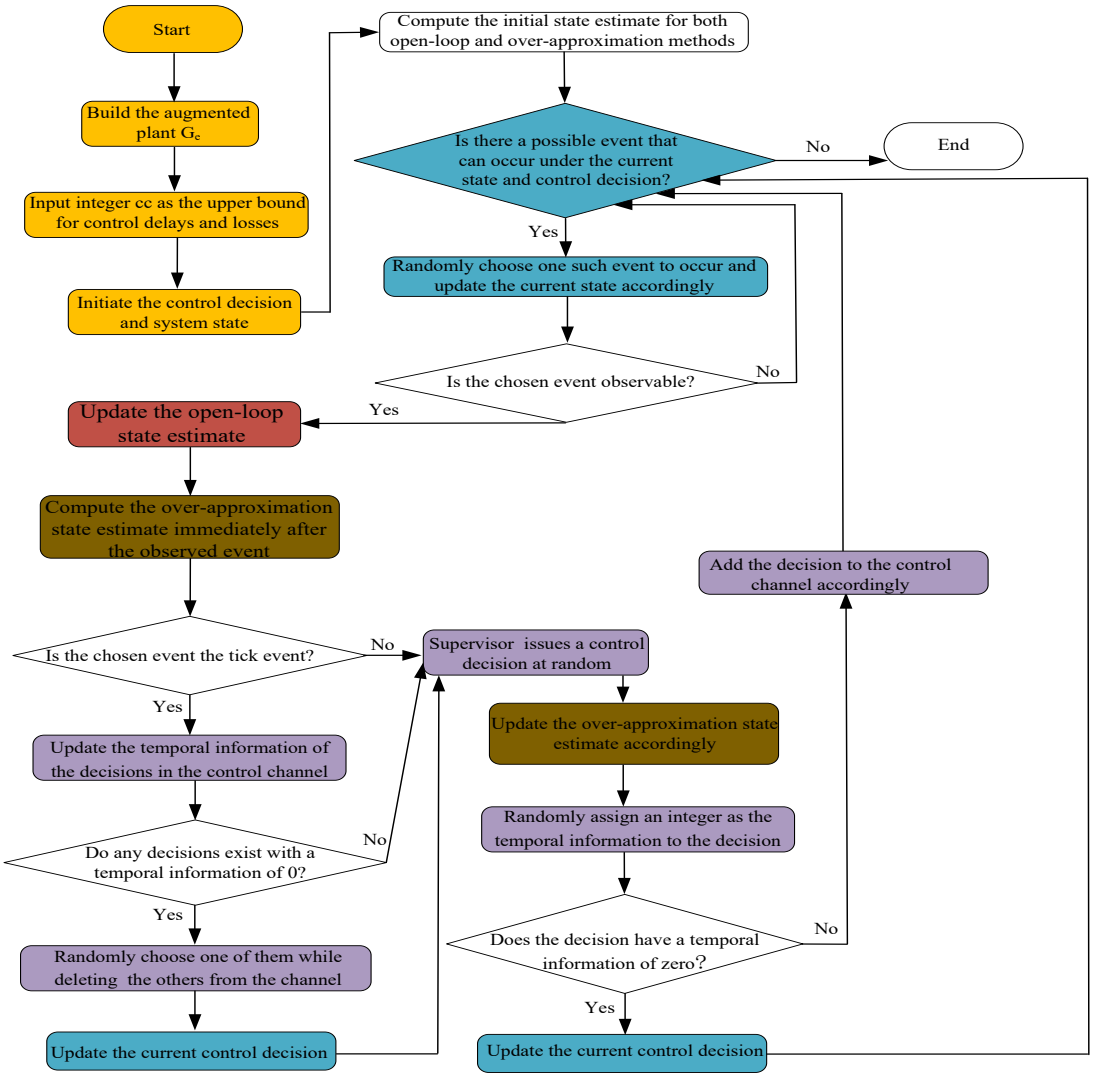

Figure 4: The algorithm flowchart for the simulation with non-FIFO control channel.

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. 5.

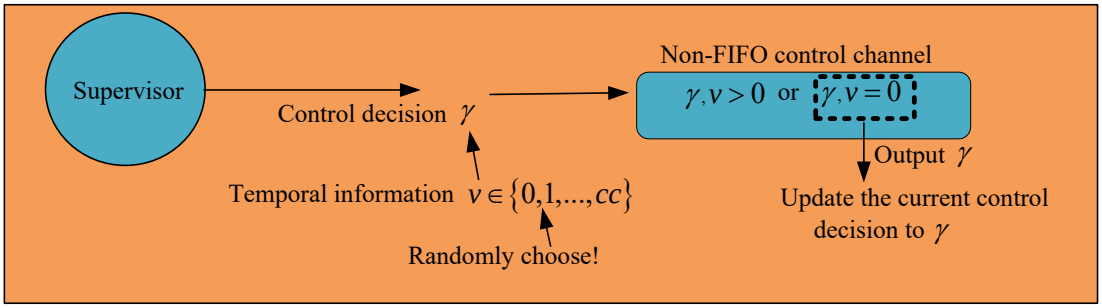

Figure 5: An illustration of modeling control delays.

(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. 6.

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

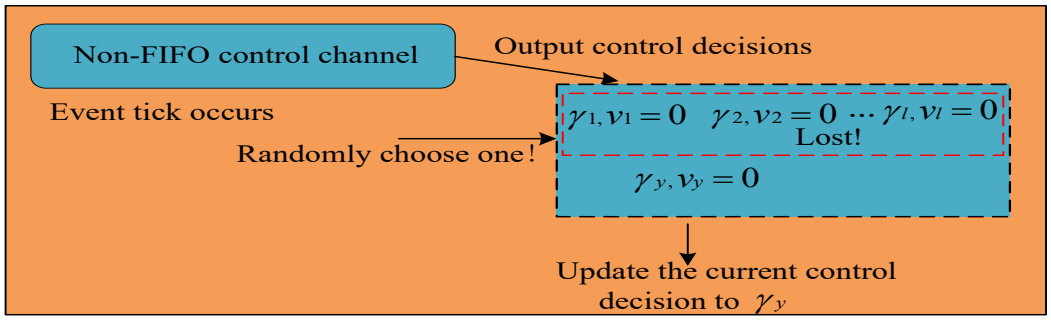

Figure 6: An illustration of modeling control losses.

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.

1.2 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. 7. 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. 4, which are for simulating a non-FIFO channel. All other steps are the same.

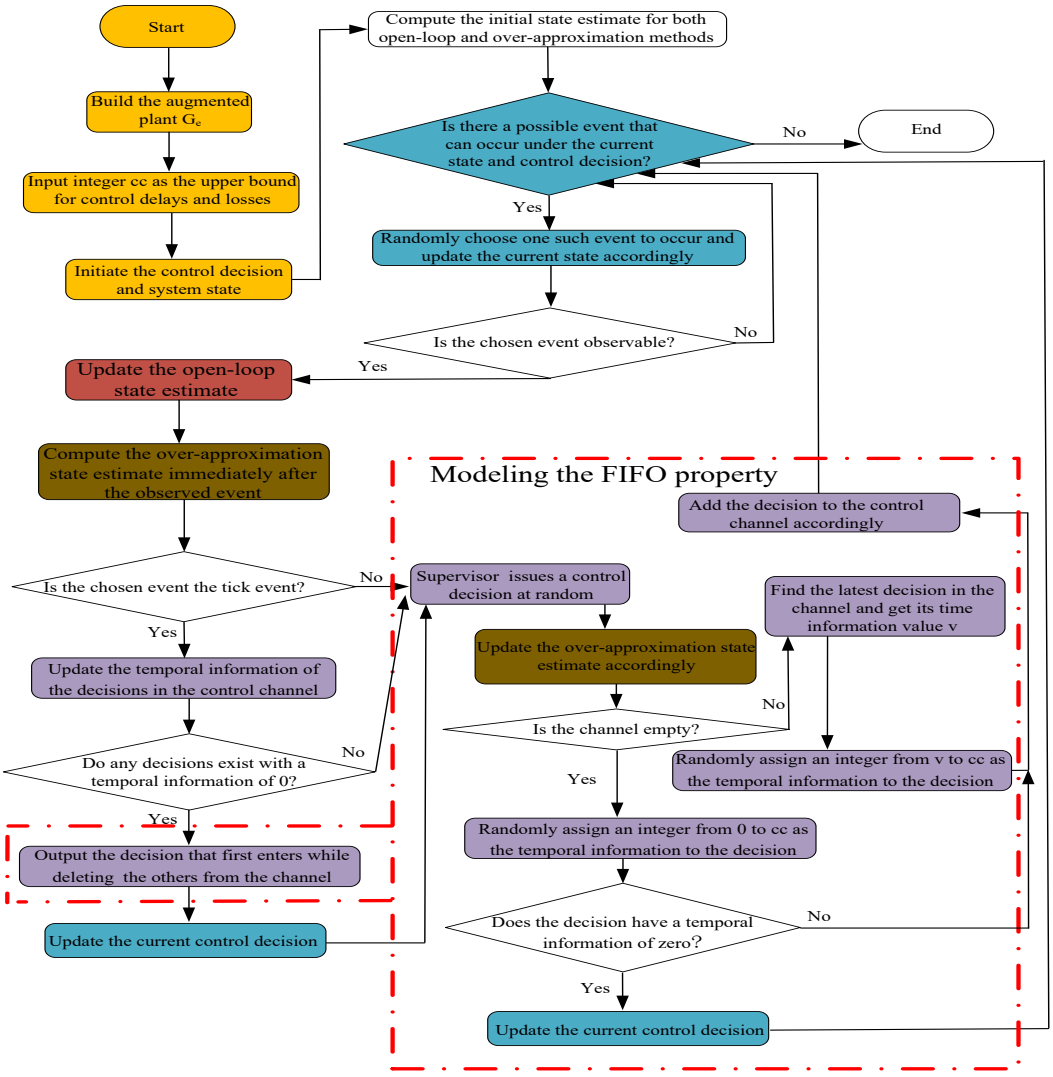

Figure 7: The algorithm flowchart for the simulation with FIFO control channel.

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. 5, while the case that the channel is not empty is shown in Fig. 8.

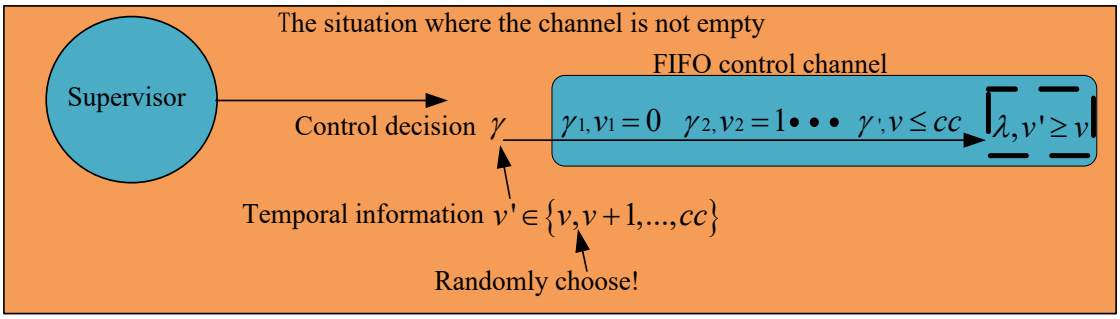

Figure 8: An illustration of modeling control delays for FIFO channel.

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. 9. We finally provide simulation data with $cc = 1$, as illustrated in Table 6

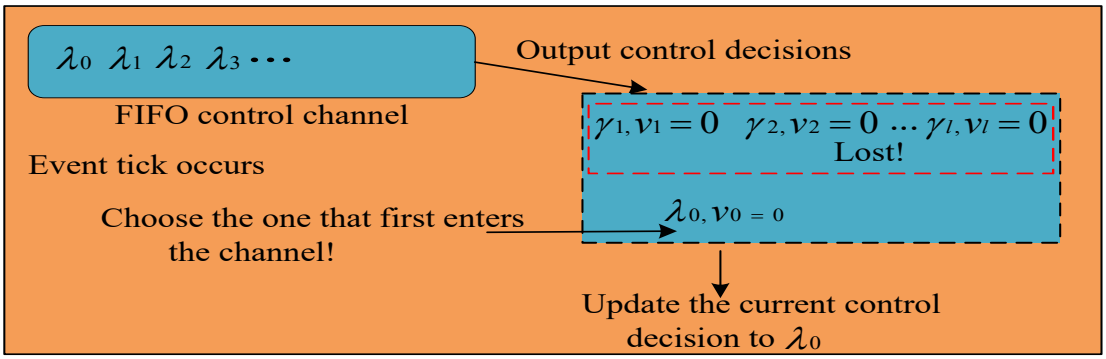

Figure 9: An illustration of modeling control losses for FIFO channel.

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}

1.3 Simulation Conclusion

Through the simulation, we have verified the applicability of the proposed over-approximation state estimation method to both non-FIFO (see 1.1) and FIFO (see 1.2) 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.