Maximally Permissive Data-Driven Supervisory Control of Discrete-Event Systems with Forcible Events—An Extended Version

Chao Gu * **Chao Gao** ** **Kai Cai** ***

* *School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK (e-mail: cgu@qub.ac.uk)*
** *Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University, Kyoto, Japan (e-mail: gao.chao.3j@kyoto-u.ac.jp)*
*** *Department of Core Informatics, Osaka Metropolitan University, Osaka, Japan (e-mail: cai@omu.ac.jp)*

Abstract: This paper studies the synthesis of maximally permissive data-driven supervisory control for structure-unknown discrete-event systems with forcible events. Consider data as: 1) a subset of possible event sequences generated by the structure-unknown plant, and 2) a subset of impossible behaviors of the system identified from pre-established knowledge. With an event-forcing mechanism, *forcible-controllability* ensures the existence of model-based forcing supervisory control. In our data-driven setting, *forcible-informativity* is a criterion for determining the forcible-controllability of a specification using only data, while *forcible-informatizability* assesses whether the data can identify the forcible-controllability of a smaller, non-empty specification. In this work, we first show that whenever forcible-informatizability is satisfied, there exists a unique non-empty supremal forcibly-informative sublanguage of the specification for maximally permissive data-driven supervisory control. To compute this supremal sublanguage and synthesize maximally permissive data-driven forcing supervisory control, we introduce the notion of a *non-forcibly informative state*. Finally, we propose an algorithm based on the model-based maximally permissive forcing supervisory control strategy to synthesize the maximally permissive data-driven forcing supervisory control.

Keywords: Discrete-event system, supervisory control, data-driven control, forcible event.

1. INTRODUCTION

Data-driven methods (Wardi et al., 2018; Farooqui et al., 2021; Cai, 2022) offer a promising alternative when discrete-event system (DES) models are partially unknown, where conventional model-based approaches (Cassandras and Lafortune, 2021; Wonham and Cai, 2019) fall short. Inspired by the concept of *data informativity* (Van Waarde et al., 2020) for data-driven analysis of continuous-time systems, the authors in (Ohtsuka et al., 2023) extend this idea to DES. They introduce a data-driven method to verify the controllability of a specification for structure-unknown DES, given two data sets: 1) D, a subset of possible behaviors of the plant; 2) D^-, a subset of impossible behaviors (either due to prior knowledge or violation of physical principles). The concept of data informativity is used to evaluate whether D and D^- are sufficient for verifying controllability.

However, the provided data may not always be informative enough for the given specification. To address this, the concept of *data informatizability* is introduced in (Ohtsuka et al., 2026) as a key property of the data D and D^- for determining the controllability of a smaller, non-empty specification. Yet, data informativity often imposes stringent requirements on the quality of the data, especially for systems with many uncontrollable events. If neither informativity nor informatizability is satisfied, it becomes impossible to design a valid data-driven supervisor with conventional control actions, with only enabling and disabling of controllable events.

The event-forcing mechanism has been well studied in model-based supervisory control (Brandin and Wonham, 1994; Miura and Takai, 2018; Rashidinejad et al., 2023; Wonham and Cai, 2019), where forcible events are introduced to preempt time elapsing. That is, the supervisor has the ability to enforce a forcible event ahead of an uncontrollable one that is otherwise enabled. In the untimed setting, a recent work (Reniers and Cai, 2024) introduces the concept of *forcible-controllability*, which characterizes the existence of a valid supervisor to enforce an imposed specification, extending the traditional controllability.

In (Gu et al., 2024, 2025), the conventional supervisor is extended with event-forcing actions (in addition to enabling/disabling), exploring new properties of data sets that allow the design of a valid data-driven supervisor. This shift brings forcible-controllability to the data-driven (structure-unknown) supervisory control setting, leading to new properties and methods for data-driven control using the event-forcing mechanism. In particular, a property of *forcible-informativity* is introduced that provides a criterion of data quality for determining the forcible-controllability of the specification, thus enabling data-driven forcing supervisory control. If forcible-informativity fails, the notion of *forcible-informatizability* is introduced to describe the ability to determine forcible-controllability for some non-empty sublanguage of the specification.

If forcible-informatizability holds, there generally exist multiple non-empty sublanguages of the specification that are forcibly-controllable. This raises the following question: *Among all non-empty, forcibly-controllable sublanguages, is there a supremal one that imposes the least restrictive conditions for data-driven forcing supervisory control? If so, how can it be designed?* In this paper, we explore the maximally permissive (least restrictive) data-driven forcing supervisory control to identify (if it exists) the largest forcibly-informative sublanguage of the specification. The contributions are twofold, as outlined below.

- First, we prove the existence of a unique supremal forcibly-informative sublanguage of the specification and introduce *non-forcibly informative states* based on a data-driven automaton for its computation.
- Second, we propose an algorithm for synthesizing the maximally permissive data-driven supervisor by leveraging a model-based forcing supervisory control strategy (Reniers and Cai, 2024), which is generally more permissive than (Ohtsuka et al., 2026).

The structure of this paper is organized as follows. Section 2 introduces the preliminaries. Section 3 reviews the data-driven setting of DES, as well as the concepts of forcible-informativity and forcible-informatizability. Section 4 shows the existence of maximally permissive data-driven supervisory control for DES with forcible events, and introduces the concept of non-forcibly-informative states. The main contribution, which involves synthesizing the maximally permissive data-driven forcing supervisory control, is presented in Section 5. Conclusions are drawn in Section 7.

2. PRELIMINARIES

A finite-state automaton is a four-tuple $\mathbf{G} = (Q, \Sigma, \delta, q_0)$, where Q is the finite state set, Σ is the finite event set, $\delta : Q \times \Sigma \to Q$ is the partial state transition function, and $q_0 \in Q$ is the initial state. A string s on Σ is a finite sequence of events from Σ. Let ϵ be the empty string and Σ^* be the set of all strings on Σ along with ϵ. Extend the transition function as $\delta : Q \times \Sigma^* \to Q$ and write $\delta(q, s)!$ to mean that $s \in \Sigma^*$ is defined at $q \in Q$. $L(\mathbf{G}) = \{s \in \Sigma^* \mid \delta(q_0, s)!\}$ is defined as the language generated by $\mathbf{G}$. For $\mathbf{G} = (Q, \Sigma, \delta, q_0)$, a subautomaton $\mathbf{G}'$ is defined as a four-tuple $\mathbf{G}' = (Q', \Sigma', \delta', q_0')$, where $Q' \subseteq Q, \Sigma' \subseteq \Sigma, \delta' = \delta|_{Q' \times \Sigma'}$, and $q_0' \in Q'$. Subautomaton is denoted by $\mathbf{G}' \sqsubseteq \mathbf{G}$. Given two languages $L_1, L_2 \subseteq \Sigma^*$, the language concatenation of L_1 and L_2, is denoted and defined as $L_1 \cdot L_2 = \{s \in \Sigma^* \mid s = s_1 \cdot s_2, s_1 \in L_1, s_2 \in L_2\}$. String s' is a prefix of string s, written $s' \in \overline{s}$, if there exists $s'' \in \Sigma^*$ such that $s's'' = s$. The prefix closure of a language L is denoted and defined as $\overline{L} = \{s \in \Sigma^* \mid (\exists s' \in \Sigma^*) ss' \in L\}$. The *length* $|s|$ of a string $s \in \Sigma^*$ is defined according to $|\epsilon| = 0; |s| = k$, if $s = \sigma_1 \cdots \sigma_k$. Define $\max\_\mathrm{len}(L) = \{|s| \mid s \in L \wedge [(\nexists s' \in L)|s'| > |s|]\}$ as the maximal length of strings in L.

For control, the event set Σ of $\mathbf{G}$ is partitioned as $\Sigma = \Sigma_c \dot\cup \Sigma_u$, where Σ_c and Σ_u are the sets of controllable and uncontrollable events, respectively. A *supervisor* (i.e., a control agent) can enable/disable only controllable events in Σ_c. A supervisor $V : L(\mathbf{G}) \to \Gamma$, where $\Gamma = \{\gamma \mid \Sigma_u \subseteq \gamma \subseteq \Sigma\}$, disables controllable events outside $V(s)$ at each s. The closed-loop language $L(V/\mathbf{G}) \subseteq L(\mathbf{G})$ is defined in the standard way (Wonham and Cai, 2019), which includes all strings in $L(\mathbf{G})$ that are not disabled under supervisory control V. A control specification $K \subseteq L(\mathbf{G})$ is a language (normally non-empty) that we intend to impose on $\mathbf{G}$ in order to achieve a specific control goal. Let $\mathbf{G}$ be a plant and $K \subseteq L(\mathbf{G})$ be a specification language. K is controllable if $\overline{K}\Sigma_{uc} \cap L(\mathbf{G}) \subseteq \overline{K}$, i.e., $(\forall s \in \Sigma^*)(\forall \sigma \in \Sigma) s \in \overline{K}, \sigma \in \Sigma_u, s\sigma \in L(\mathbf{G}) \Rightarrow s\sigma \in \overline{K}$. Controllability (Wonham and Cai, 2019) relates to the existence of a supervisory control enforcing the specification.

Given a set of forcible events $\Sigma_{for} \subseteq \Sigma$, a forcible event $f \in \Sigma_{for}$, if it occurs at a state, preempts the occurrence of all non-forcible events at that state. The concept of *forcible-controllability* is introduced in (Reniers and Cai, 2024), which generalizes conventional controllability by allowing using forcible events to preempt uncontrollable ones.

Definition 1. *Consider a plant $\mathbf{G}$ with $\Sigma = \Sigma_c \dot\cup \Sigma_u$ and $\Sigma_{for} \subseteq \Sigma$. A control specification $K \subseteq L(\mathbf{G})$ is forcibly-controllable with respect to $\mathbf{G}$ if:*

$$(\forall s \in \overline{K}, \forall \sigma \in \Sigma_u)\ s\sigma \in L(\mathbf{G}) \Rightarrow [s\sigma \in \overline{K}] \vee$$
$$[((\exists f \in \Sigma_{for}) sf \in \overline{K}) \wedge ((\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K})].$$

Controllability implies forcible-controllability but not vice versa. Specifically, controllability involves only the ability to disable events, whereas forcible-controllability includes both disabling and forcing. In other words, forcible-controllability allows potential violations of controllability to be preempted by the use of appropriate forcible events.

(Reniers and Cai, 2024, Theorem 1) shows that the forcible-controllability of K ensures the existence of a supervisory control with event-forcing to enforce K. In particular, if controllability is already satisfied, no event-forcing is necessary. However, if controllability is violated, certain forcible events are used to preempt events that violate controllability. Further, the set of *forcibly-controllable sublanguages* for the specification K is denoted as $\mathcal{F}(K) = \{K' \subseteq K \mid K' \text{ is forcibly-controllable with respect to } \mathbf{G}\}$. Forcible-controllability is proved to be closed under union and thus the set $\mathcal{F}(K)$ contains a unique supremal element $\sup\mathcal{F}(K) = \bigcup_{K' \in \mathcal{F}(K)} K'$. If $\sup\mathcal{F}(K) \neq \emptyset$, there exists a non-empty maximally permissive supervisor to enforce $\sup\mathcal{F}(K)$, which can be synthesized as detailed in (Reniers and Cai, 2024, Algorithm 1).

3. FORCIBLE-INFORMATIVITY AND FORCIBLE-INFORMATIZABILITY

This section reviews two key concepts related to data-driven supervisory control of DESs with forcible events: forcible-informativity and forcible-informatizability (Gu et al., 2024, 2025). First, the data-driven setting is presented in Assumption 1 (Gu et al., 2025).

Assumption 1. *Consider a plant $\mathbf{G} = (Q, \Sigma, \delta, q_0)$ whose state set Q, transition function δ, and initial state q_0 are unknown. We assume that the following are known: (i) Event set Σ, the controllable event set $\Sigma_c \subseteq \Sigma$, and the forcible event set $\Sigma_{for} \subseteq \Sigma$; (ii) A specification (regular) language $E \subseteq \Sigma^*$; (iii) A subset of $\mathbf{G}$'s behavior is known, which is denoted as a finite data set $D \subseteq L(\mathbf{G})$; (iv) Another (possibly infinite and regular) subset $D^- \subseteq \Sigma^*$ is also known, consisting of strings that can never happen in $\mathbf{G}$, i.e., $D^- \subseteq \Sigma^* \setminus L(\mathbf{G})$.*

The data set D can be collected by logging or observing the unknown plant, while D^- can be derived in practical scenarios from prior knowledge, such as sequences that violate physical laws or specified requirements[1]. Both data sets are assumed to be fixed. Note that from $D \subseteq L(\mathbf{G})$ and $D^- \subseteq \Sigma^* \setminus L(\mathbf{G})$, we have that $D \cap D^- = \emptyset$. The data sets D, D^- may correspond to multiple plant models that can generate the behaviors in D and no behavior in D^-. A plant $\mathbf{G}$ is said to be *consistent* with the data pair (D, D^-) if $D \subseteq L(\mathbf{G})$ and $D^- \cap L(\mathbf{G}) = \emptyset$. Besides, the specification with regard to the data is denoted as

$$D_E = \overline{D} \cap E. \tag{1}$$

By Assumption 1 (for finite D^-), a key property called informativity is introduced in (Ohtsuka et al., 2023): a finite data pair (D, D^-) is informative for a specification E if D_E in (1) is non-empty and controllable with respect to all plants consistent with (D, D^-). This enables data-driven supervisory control for enforcing D_E across all plants consistent with (D, D^-), including the true plant. However, ensuring informativity by only disabling/enabling controllable events can place stringent requirements on the quality of the data (Gu et al., 2025). To alleviate these data requirements, the additional control mechanism, event-forcing, is integrated, leading to the introduction of the concept of forcible-informativity (Gu et al., 2024, 2025), as defined below.

Definition 2. *Consider $\Sigma = \Sigma_c \dot\cup \Sigma_u$ with $\Sigma_{for} \subseteq \Sigma$ and a specification $E \subseteq \Sigma^*$. Given $D, D^- \subseteq \Sigma^*$, (D, D^-) is said to be forcibly-informative for E if the non-empty specification D_E is forcibly-controllable for all plants consistent with (D, D^-), i.e., there exists a supervisory control for all data consistent plants to enforce D_E.*

More general than informativity in (Ohtsuka et al., 2023), forcible-informativity indicates that a supervisory control can be synthesized using an event-forcing mechanism to enforce the non-empty specification D_E for any plant consistent with (D, D^-), relying only on the available data. A necessary and sufficient condition for verifying forcible-informativity is provided in Proposition 1 (Gu et al., 2024, 2025).

Proposition 1. *Consider an event set $\Sigma = \Sigma_c \dot\cup \Sigma_u$ with $\Sigma_{for} \subseteq \Sigma$ and a specification $E \subseteq \Sigma^*$. Given $D, D^- \subseteq \Sigma^*$ with $D_E = \overline{D} \cap E$, the pair (D, D^-) is forcibly-informative for specification E if and only if the following holds:*

$$(\forall s \in \overline{D_E}, \forall \sigma \in \Sigma_u)[s\sigma \in \overline{D_E} \cup D^-] \vee$$
$$[((\exists f \in \Sigma_{for}) sf \in \overline{D_E}) \wedge ((\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{D_E})].$$

Proposition 1 suggests that forcible-informativity can be verified by examining all strings in the specification $\overline{D_E}$ and all uncontrollable events in Σ_u.

To verify forcible-informativity algorithmically, the concept of a *data-driven automaton* is introduced in (Gu et al., 2025) (see Definition 3 below), which is based on a finite-state automaton representing $D^- \subseteq \Sigma^*$, denoted as $\mathbf{G}_{\mathbf{D}^-} = (Q^-, \Sigma, \delta^-, q_0^-)$ (Gu et al., 2025). This is always possible since, by Assumption 1, D^- is a regular language. Note that $L(\mathbf{G}_{\mathbf{D}^-}) = \overline{D^-}$. Meanwhile, we define $Q_{D^-} = \{\delta^-(q_0^-, s) \mid s \in D^-\} \subseteq Q^-$, denoting all states corresponding to strings in D^-.

Definition 3. *Consider an event set Σ, a specification $E \subseteq \Sigma^*$, and data $D, D^- \subseteq \Sigma^*$ with an automaton $\mathbf{G}_{\mathbf{D}^-}$ and set Q_{D^-} representing D^-. A data-driven automaton is defined as $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat{\delta}, \hat{q}_0)$, where $\hat{Q} = Q^- \cup \{\hat{q}_s \mid s \in \overline{D}\}$ is the state set where $\hat{q}_\epsilon = q_0^-$, $\hat{\delta} = \delta^- \cup \{(\hat{q}_s, \sigma) \to \hat{q}_{s\sigma} \mid (\sigma \in \Sigma) \wedge (s, s\sigma \in \overline{D})\}$ is the state transition function, and $\hat{q}_0 = q_0^-$ is the initial state.*

A data-driven automaton is a finite-state automaton where $L(\hat{\mathbf{G}}(\Sigma, E, D, D^-)) = \overline{D} \cup \overline{D^-}$ with $D^- \subseteq L(\hat{\mathbf{G}}(\Sigma, E, D, D^-))$ corresponding to the subset $Q_{D^-} \subseteq Q^- \subseteq \hat{Q}$. The set $\hat{Q}_{D_E} = \{\hat{q} \in \hat{Q} \mid (s \in \overline{D_E}) \wedge (\hat{q}_0, s) \to \hat{q}\}$ is defined as the subset of states in $\hat{Q}$ that correspond to strings in $\overline{D_E}$. Using the data-driven automaton, (Gu et al., 2025, Algorithm 1) is introduced to verify forcible-informativity. The construction of a data-driven automaton has a complexity $O(|D| \cdot \text{max\_len}(D) + |Q^-|)$. If D^- is finite, this simplifies to $O(|D \cup D^-| \cdot \text{max\_len}(D \cup D^-))$.

Under the condition tha the data pair (D, D^-) is forcibly-informative for the specification E, by (Gu et al., 2025), for any plant $\mathbf{G}$ consistent with (D, D^-), a data-driven supervisory control $V_{D,for} : \overline{D} \to 2^\Sigma$ can be designed such that $L(V_{D,for}/\mathbf{G}) = \overline{D_E}$, according to

$$V_{D,for}(s) := \begin{cases} \Sigma_u \dot\cup \{t \in \Sigma_c \mid st \in \overline{D_E}\}, \\ \quad \text{if } [s \in \overline{D_E}] \wedge [(\forall \sigma \in \Sigma_u)\ s\sigma \in \overline{D_E} \cup D^-]; \\ \{f \in \Sigma_{for} \mid sf \in \overline{D_E}\}, \\ \quad \text{if } [s \in \overline{D_E}] \wedge [(\exists \sigma' \in \Sigma_u)\ s\sigma' \notin \overline{D_E} \cup D^-]; \\ \Sigma, \ \text{if } s \in \overline{D} \setminus \overline{D_E}. \end{cases}$$

When the data pair (D, D^-) is not forcibly-informative for the specification E (as defined in Definition 2), synthesizing a data-driven supervisor to enforce the non-empty specification D_E becomes impossible. To explore the possibility of enforcing a smaller non-empty specification, the concept of *forcible-informatizability* is introduced and defined below (Gu et al., 2025).

Definition 4. *Consider $\Sigma = \Sigma_c \dot\cup \Sigma_u$ with $\Sigma_{for} \subseteq \Sigma$. Given a pair (D, D^-) and a specification language $E \subseteq \Sigma^*$ with $D_E = \overline{D} \cap E$, the pair (D, D^-) is said to be forcibly-informatizable for E if there exists a non-empty language $K \subseteq \overline{D_E}$ such that K is forcibly-controllable with respect to every plant $\mathbf{G}$ consistent with (D, D^-), i.e., there exists a supervisory control for $\mathbf{G}$ to enforce K.*

The concept of forcible-informatizability shows that even if the data pair (D, D^-) is not forcibly-informative for E, a data-driven forcing supervisor can still be designed to enforce a smaller non-empty sublanguage $K \subseteq \overline{D_E}$. A necessary and sufficient condition for verifying forcible-informatizability is proposed in (Gu et al., 2025, Theorem 1), along with a data-driven automaton-based algorithm for its implementation. Corollary 1 illustrates the relationship between forcible-informatizability and forcible-informativity with respect to the specification sublanguage $K \subseteq \overline{D_E}$.

Corollary 1. *A data pair (D, D^-) is forcibly-informatizable for a specification E if and only if there exists a non-empty specification language $K \subseteq \overline{D_E}$ such that (D, D^-) is forcibly-informative for K.*

Example 1. *Consider a robot exploring an unknown environment and collecting data. $\Sigma = \{u_1, u_2, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$ denotes the set of events representing different robot movements. The set $\Sigma_u = \{u_1, u_2\}$ contains uncontrollable movements (e.g., motions caused by disturbances), while $\Sigma_c = \Sigma_{for} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$ represents controllable movements (e.g., intended actions)*

[1] This paper is based on the assumption that D and D^- are given and correct. A systematic study of the practical acquisition of D and D^- in real-world scenarios, and even partially incorrect (e.g., noisy data) D and D^-, will be explored in the future.

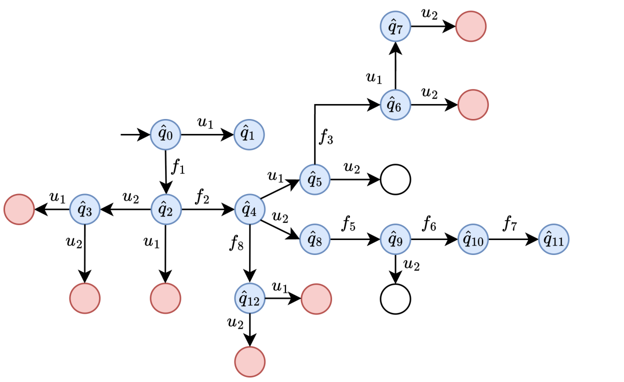

Fig. 1. A data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-)$ with states in $\hat{Q}_{D_E}$ in blue and Q_{D^-} in red.

that can also preempt non-forcible ones by being executed faster. The specification $E = \{u_1, f_1u_2, f_1f_2u_1f_3u_1, f_1f_2f_8, f_1f_2u_2f_5f_6f_7\}$ *defines five target navigation routes of the robot. Given an observed data log* $D = \{u_1, f_1f_2u_2f_5f_6f_7, f_1u_2, f_1f_2u_1u_2, f_1f_2u_1f_3u_1, f_1f_2f_8, f_1f_2u_2f_5u_2\}$*, collected via robot movement observations, and a set of impossible behaviors* $D^- = \{f_1u_1, f_1u_2u_1, f_1u_2u_2, f_1f_2u_1f_3u_2, f_1f_2f_8u_1, f_1f_2f_8u_2, f_1f_2u_1f_3u_1u_2\}$*, constructed based on prior knowledge or robot operation documentation.*

We verify the lack of forcible-informativity of (D, D^-) *for* E *by Proposition 1. Consider a string* $u_1 \in \overline{D_E}$ *and an uncontrollable event* $u_1 \in \Sigma_u$*. Since* $u_1u_1 \notin \overline{D_E} \cup D^-$ *and there does not exist any forcible event* $f \in \Sigma_{for}$ *such that* $u_1f \in \overline{D_E}$*, we conclude that the data pair* (D, D^-) *is not forcibly-informative for* E*. Note that this can be verified via (Gu et al., 2025, Algorithm 1) based on a data-driven automaton* $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat{\delta}, \hat{q}_0)$ *in Fig. 1, with state set* $\hat{Q}_{D_E} = \{\hat{q}_0, \ldots, \hat{q}_{12}\}$ *(color-coded in blue) and* Q_{D^-} *(not numbered for simplicity, color-coded in red).*

Additionally, the forcible-informatizability of (D, D^-) *can be verified using the data-driven automaton through (Gu et al., 2025, Algorithm 2). For example, consider a non-empty language* $K = \{f_1u_2\} \subseteq \overline{D_E}$*. Since* (D, D^-) *can be verified as forcibly-informative for* K*, it follows from Corollary 1 that* (D, D^-) *is forcibly-informatizable for* E*.*

4. EXISTENCE OF MAXIMALLY PERMISSIVE DATA-DRIVEN SUPERVISORY CONTROL WITH FORCIBLE EVENTS

If forcible-informatizability holds, it is reasonable to identify a sublanguage $K \subseteq \overline{D_E}$ such that (D, D^-) is forcibly-informative for K, while imposing *minimal restrictions* on the structure-unknown plant. Specifically, the goal is to determine a sublanguage of $\overline{D_E}$ that maximizes behavioral permissiveness when implementing data-driven forcing supervisory control.

This section establishes the existence of an *maximally permissive* data-driven forcing supervisory control in the sense that minimal restriction is imposed on the unknown plant. Using the data-driven automaton, a set of states called *non-forcibly-informative states* is introduced, along with an algorithm for their computation. These states correspond to strings in the specification $\overline{D_E}$ that violate both controllability and forcible-controllability, which are key to the design of data-driven forcing supervisory control. First, an illustrative example is provided below.

Example 2. *Revisit the example in Example 1. The data pair* (D, D^-) *has been verified to be forcibly-informatizable for the specification* E*. Consider another sublanguage* $K' = \{f_1u_2, f_1f_2f_8\} \subseteq \overline{D_E}$*. By Proposition 1, we can verify that* (D, D^-) *is also forcibly-informative for* K'*. Comparing* $K = \{f_1u_2\}$ *and* $K' = \{f_1u_2, f_1f_2f_8\}$*, it is clear that* K' *imposes fewer restrictions on the specification* $\overline{D_E}$ *than* K *does. When enforcing* K'*, strings* $f_1f_2, f_1f_2f_8$ *will not be discarded during data-driven forcing supervisory control.*

Definition 5. *For the given data* (D, D^-) *and the specification* E*, we define a class of sublanguages* $\mathcal{I}_{for}(D, E) = \{K \subseteq \overline{D_E} \mid K \neq \emptyset \wedge (D, D^-)$ *is forcibly-informative for* $K\}$*.*

From Definition 5, the set $\mathcal{I}_{for}(D, E)$ contains a non-empty element if (D, D^-) is forcibly-informatizable (by Corollary 1). Moreover, $\mathcal{I}_{for}(D, E)$ is finite because the data set D is finite, as stated in Assumption 1. Proposition 2 below shows $\mathcal{I}_{for}(D, E)$ is closed under set unions.

Proposition 2. *If* $K_1, K_2 \in \mathcal{I}_{for}(D, E)$*, then* $K_1 \cup K_2 \in \mathcal{I}_{for}(D, E)$*.*

Proof. Consider a string $s \in \overline{K_1 \cup K_2}$ and an uncontrollable event $\sigma \in \Sigma_u$. Since (D, D^-) is forcibly-informative for $K_1 \cup K_2$, we prove that

$$[s\sigma \in \overline{K_1 \cup K_2} \cup D^-] \vee \\ [((\exists f \in \Sigma_{for})\ sf \in \overline{K_1 \cup K_2}) \wedge \\ ((\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K_1 \cup K_2}]. \tag{2}$$

Since $s \in \overline{K_1 \cup K_2} = \overline{K_1} \cup \overline{K_2}$, either $s \in \overline{K_1}$ or $s \in \overline{K_2}$ holds. Assume that $s \in \overline{K_1}$ (analogously for $s \in \overline{K_2}$ hereinafter). Since (D, D^-) is forcibly-informative for K_1, either $s\sigma \in \overline{K_1} \cup D^- \subseteq \overline{K_1} \cup \overline{K_2} \cup D^- = \overline{K_1 \cup K_2} \cup D^-$, or

$$(\exists f \in \Sigma_{for})\ sf \in \overline{K_1} \subseteq \overline{K_1} \cup \overline{K_2} = \overline{K_1 \cup K_2} \tag{3}$$

and

$$(\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K_1}. \tag{4}$$

Following (4), there are two sub-cases: either (i) $(\forall \sigma'' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K_2}$, or (ii) $(\exists \sigma'' \in \Sigma \setminus \Sigma_{for})\ s\sigma'' \in \overline{K_2}$. For the former, we have $s\sigma' \notin \overline{K_1} \cup \overline{K_2} = \overline{K_1 \cup K_2}$. For the latter, since $s \in \overline{K_2}$, $s \in \overline{K_1} \cap \overline{K_2}$ holds. We analyze the latter case in detail below.

Since (D, D^-) is forcibly-informative for K_2, for $s \in \overline{K_2}$ and any $u \in \Sigma_u$, either $su \in \overline{K_2} \cup D^- \subseteq \overline{K_1} \cup \overline{K_2} \cup D^- = \overline{K_1 \cup K_2} \cup D^-$, or

$$(\exists f' \in \Sigma_{for})\ sf' \in \overline{K_2} \subseteq \overline{K_1} \cup \overline{K_2} = \overline{K_1 \cup K_2} \tag{5}$$

and

$$(\forall t \in \Sigma \setminus \Sigma_{for}) st \notin \overline{K_2}. \tag{6}$$

Note that (6) contradicts the previous subcase (ii). Therefore, for subcase (ii), only $su \in \overline{K_1 \cup K_2} \cup D^-$ holds, which satisfies (2) and concludes the proof. $\square$

By Proposition 2, for a data pair (D, D^-) with a specification E, the set $\mathcal{I}_{for}(D, E)$ has a unique *supremal* element (if it is not empty), namely the *supremal forcibly-informative sublanguage* of D_E, denoted as:

$$K^{\uparrow I} := \bigcup_{K \in \mathcal{I}_{for}(D,E)} K.$$

Furthermore, it follows that $K^{\uparrow I} = \overline{D_E}$ if (D, D^-) is forcibly-informative. By Definition 2, $K^{\uparrow I}$ is forcibly-controllable with respect to any plant $\mathbf{G}$ consistent with (D, D^-). In other words, there exists a supervisory control $V_{K^{\uparrow I}}$ for $\mathbf{G}$ such that $L(V_{K^{\uparrow I}}/\mathbf{G}) = K^{\uparrow I}$. This supervisory control, $V_{K^{\uparrow I}}$, is the maximally permissive data-driven forcing supervisory control because it imposes the minimal restrictions on any $\mathbf{G}$ consistent with (D, D^-).

In the next section we will show how to synthesize the maximally permissive data-driven forcing supervisory $V_{K^{\uparrow I}}$. To achieve this, we introduce the concept of *non-forcibly-informative states* with respect to a data-driven automaton. This concept serves as the foundation for the synthesis process.

Definition 6. *Consider $\Sigma = \Sigma_c \dot\cup \Sigma_u, \Sigma_{for} \subseteq \Sigma$, the specification $E \subseteq \Sigma^*$, and a data pair (D, D^-). For a data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat\delta, \hat q_0)$ with $\hat{Q}_{D_E}$ and Q_{D^-}, the set of non-forcibly-informative states is defined as $\hat{Q}_{nf} = \{q \in \hat{Q}_{D_E} \mid (\exists \sigma \in \Sigma_u)[\hat\delta(q, \sigma) \notin \hat{Q}_{D_E} \cup Q_{D^-}] \wedge [(\nexists f \in \Sigma_{for})\hat\delta(q, f) \in \hat{Q}_{D_E}]\}$.*

According to Definition 6, $\hat{Q}_{nf} \subseteq \hat{Q}_{D_E}$ holds. Additionally, the strings in $\overline{D_E}$ corresponding to the non-forcibly-informative states in $\hat{Q}_{nf}$ do not satisfy part of the criteria for forcible-informativity outlined in Proposition 1. Therefore, (D, D^-) is not forcibly-informative if $\hat{Q}_{nf} \neq \emptyset$ holds. In other words, $\hat{Q}_{nf} = \emptyset$ if the data pair (D, D^-) is forcibly-informative for the specification E. An example illustrating this is provided below.

Example 3. *Revisit the example in Example 1. The data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat\delta, \hat q_0)$, with the state set $\hat{Q}_{D_E}$ is illustrated in Fig. 1. Consider a state $\hat q_1 \in \hat{Q}_{D_E}$ and an event $u_1 \in \Sigma_u$. Since $\hat\delta(\hat q_1, \sigma) \notin \hat{Q}_{D_E} \cup Q_{D^-}$ and there does not exist any forcible event $f \in \Sigma_{for}$ such that $\hat\delta(\hat q_1, f) \in \hat{Q}_{D_E}$, we conclude that $\hat q_1 \in \hat{Q}_{nf}$. Similarly, it can be determined that $\hat{Q}_{nf} = \{\hat q_1, \hat q_7, \hat q_{11}\}$. Additionally, the data pair (D, D^-) is not forcibly-informative for E because $\hat{Q}_{nf} \neq \emptyset$.*

Algorithmically, the set $\hat{Q}_{nf}$ can be obtained by using the data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat\delta, \hat q_0)$ with $\hat{Q}_{D_E}$ and Q_{D^-}, directly following Definition 6. Specifically, a state $q \in \hat{Q}_{D_E}$ also belongs to $\hat{Q}_{nf}$ if the following holds (corresponding to the condition in Definition 6):

$$[(\exists \sigma_u \in \Sigma_u)(\hat\delta(q, \sigma_u) \notin \hat{Q}_{D_E} \cup Q_{D^-}) \vee (\neg\hat\delta(q, \sigma_u)!)] \\ \wedge [(\nexists f \in \Sigma_{for})\hat\delta(q, f) \in \hat{Q}_{D_E}]$$

The computational complexity of the above acquisition of $\hat{Q}_{nf}$ primarily depends on the construction of the data-driven automaton, as discussed earlier in Section 3. We will demonstrate in the next section how non-forcibly-informative states can be used to achieve maximally permissive data-driven forcing supervisory control.

5. SYNTHESIS OF MAXIMALLY PERMISSIVE DATA-DRIVEN SUPERVISORY CONTROL WITH FORCIBLE EVENTS

This section shows how to obtain the maximally permissive data-driven forcing supervisor. Specifically, by computing the non-forcibly-informative states and leveraging the model-based supervisory control with event-forcing (Reniers and Cai, 2024), an algorithm is proposed for synthesizing a supervisor that enforces the unique supremal forcibly-informative sublanguage of $\overline{D_E}$, i.e., $K^{\uparrow I} \in \mathcal{I}_{for}(D, E)$, for a given data pair (D, D^-) with a specification E. Algorithm 1 below presents the process for obtaining a supervisor $V_{K^{\uparrow I}}$ such that $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$. A detailed explanation of the steps is given below.

Initialization (lines 1 and 2): First, a data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-)$ is constructed and the non-forcibly-informative state set $\hat{Q}_{nf}$ is computed;

Introducing a virtual uncontrollable event (line 3): A new virtual uncontrollable event, σ_u^+, is introduced. The set of uncontrollable events is then updated to $\Sigma_u^+ = \Sigma_u \cup \{\sigma_u^+\}$. The event set is expanded to $\Sigma^+ = \Sigma \cup \{\sigma_u^+\}$ to prepare for updating the data-driven automaton. The transition function is reinitialized as $\hat\delta^+ = \hat\delta$, and the state set is redefined as $\hat{Q}^+ = \hat{Q}$. The reason that we bring in σ_u^+ is, although our setting is data-driven and the plant structure is unknown, the subsequent synthesis leverages the model-based event-forcing framework of (Reniers and Cai, 2024). The additional event σ_u^+ ensures that the model-based results can be correctly applied. Further details are provided in the proof of correctness of Algorithm 1, i.e., Theorem 1;

Updating the data-driven automaton (lines 4−10): For any state q in $\hat{Q}_{D_E}$ that is *not* non-forcibly informative, if at least one forcible event $f \in \Sigma_{for}$ is needed to preempt the occurrence of an uncontrollable event σ_u (because the informativity of (D, D^-) fails due to σ_u),

---

Algorithm 1 Synthesis of the maximally permissive data-driven forcing supervisor $V_{K^{\uparrow I}}$

Require: $\Sigma = \Sigma_c \dot\cup \Sigma_u, \Sigma_{for} \subseteq \Sigma$, the specification $E \subseteq \Sigma^*$, $D \subseteq \Sigma^*$ and $D^- \subseteq \Sigma^*$
Ensure: The supervisor $V_{K^{\uparrow I}}$ where $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$

1: Construct a data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-) = (\hat{Q}, \Sigma, \hat\delta, \hat q_0)$ with $\hat{Q}_{D_E} \subseteq \hat{Q}$ and $Q_{D^-} \subseteq \hat{Q}$
2: Compute the set of non-forcibly-informative states $\hat{Q}_{nf}$ based on $\hat{\mathbf{G}}(\Sigma, E, D, D^-)$
3: $\Sigma^+ := \Sigma \cup \{\sigma_u^+\}$ with $\Sigma_u^+ := \Sigma_u \cup \{\sigma_u^+\}$; $\hat\delta^+ := \hat\delta$; $\hat{Q}^+ := \hat{Q}$
4: **for all** $q \in \hat{Q}_{D_E} \setminus \hat{Q}_{nf}$ **do**
5:     **if** $[(\exists \sigma_u \in \Sigma_u)(\hat\delta(q, \sigma_u) \notin \hat{Q}_{D_E} \cup Q_{D^-}) \vee (\neg\hat\delta(q, \sigma_u)!)] \wedge [(\exists f \in \Sigma_{for})\hat\delta(q, f) \in \hat{Q}_{D_E}]$ **then**
6:         $\hat\delta^+ := \hat\delta^+ \cup \{(q, \sigma_u^+) \to q^+\}$
7:         $\hat{Q}^+ := \hat{Q}^+ \cup \{q^+\}$
8:     **end if**
9: **end for**
10: $\hat{\mathbf{G}}^+ := (\hat{Q}^+, \Sigma^+, \hat\delta^+, \hat q_0)$
11: Construct the subautomaton $\hat{\mathbf{G}}_{plant} = (\hat{Q}', \Sigma^+, \hat\delta', \hat q_0) \sqsubseteq \hat{\mathbf{G}}^+$ where $\hat{Q}' = \hat{Q}_{D_E} \cup \{q^+ \in \hat{Q}^+ \mid (\exists q \in \hat{Q}_{D_E}, \exists \sigma_u^+ \in \Sigma_u^+)\hat\delta(q, \sigma_u^+) = q^+\}$
12: Construct the subautomaton $\hat{\mathbf{G}}_{spec} = (\hat{Q}'', \Sigma^+, \hat\delta'', \hat q_0) \sqsubseteq \hat{\mathbf{G}}_{plant}$ where $\hat{Q}'' = \hat{Q}_{D_E} \setminus \hat{Q}_{nf}$
13: Compute the maximally permissive, forcibly-controllable supervisor $V_{K^{\uparrow I}}$ by (Reniers and Cai, 2024, Algorithm 1) with $\Sigma_u^+ \subseteq \Sigma^+, \Sigma_{for} \subseteq \Sigma$, $\hat{\mathbf{G}}_{plant}$ as the plant, and $\hat{\mathbf{G}}_{spec}$ as the specification
14: **return** $V_{K^{\uparrow I}}$

---

an additional state q^+ is added to the automaton. A corresponding transition $(q, \sigma_u^+) \to q^+$ is also included. Following this process, in line 10, an updated automaton, denoted as $\hat{\mathbf{G}}^+ = (\hat{Q}^+, \Sigma^+, \hat{\delta}^+, \hat{q}_0)$, is constructed based on the modified data-driven automaton The augmented states q^+ derived from $q \in \hat{Q}_{D_E}$ serve as "flag states" that assist in trimming potentially undesirable states q which may be omitted by the subsequently adopted model-based forcing supervisory control algorithm without q^+, since the structure of the true plant $\mathbf{G}$ is unknown;

Synthesis of the maximally permissive data-driven forcing supervisor (lines 11–14): A plant model $\hat{\mathbf{G}}_{plant}$ (a subautomaton of $\hat{\mathbf{G}}^+$) and a specification $\hat{\mathbf{G}}_{spec}$ (also a subautomaton of $\hat{\mathbf{G}}^+$) are constructed. The maximally permissive data-driven forcing supervisor, which is both maximally permissive and forcibly-controllable, is then synthesized using the model-based maximally permissive forcing supervisory control approach presented in (Reniers and Cai, 2024).

Note that the virtual uncontrollable event σ_u^+ is not part of the event set Σ for the structure-unknown plant $\mathbf{G}$ in our Assumption 1. However, by introducing σ_u^+, we will demonstrate that the task of obtaining an maximally permissive data-driven forcing supervisor $V_{K^{\uparrow I}}$ where $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$ can be reformulated as a model-based maximally permissive forcing supervisory control problem demonstrated in (Reniers and Cai, 2024). The correctness of Algorithm 1 is established in Theorem 1, which is the main result of our paper.

Theorem 1. *For a data pair (D, D^-) with a specification E, the supervisor $V_{K^{\uparrow I}}$ returned by Algorithm 1 satisfies $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$.*

We analyze the complexity of Algorithm 1. First, lines 1–2 depend on the construction of a data-driven automaton, which has the complexity of $O(|D| \cdot \text{max\_len}(D) + |Q^-|)$. If D^- is finite, this simplifies to $O(|D \cup D^-| \cdot \text{max\_len}(D \cup D^-))$. From lines 3–10, within the new automaton $\hat{\mathbf{G}}^+$, at most $|\hat{Q}^+| = 2 \cdot |\hat{Q}|$ holds. The constructions of $\hat{\mathbf{G}}_{plant}$ and $\hat{\mathbf{G}}_{spec}$ on lines 11 and 12 have the same complexity of automaton $\hat{\mathbf{G}}^+$. According to (Reniers and Cai, 2024, Theorem 5), the supervisory control stage at line 13 has a complexity of $O((2 \cdot |\hat{Q}|)^2 |\Sigma|)$. In summary, Algorithm 1 has a complexity of $O((|D| \cdot \text{max\_len}(D) + |Q^-|)^2 |\Sigma| + |D| \cdot \text{max\_len}(D) + |Q^-|)$, which can be simplified to $O(|D \cup D^-|^2 (\text{max\_len}(D \cup D^-))^2 |\Sigma| + |D \cup D^-| \cdot \text{max\_len}(D \cup D^-))$ if D^- is finite. An example of synthesizing the maximally permissive data-driven forcing supervisor is shown below.

Example 4. *[Example 3 ext.] Consider the data-driven automaton in Fig. 1 with $\hat{Q}_{nf} = \{\hat{q}_1, \hat{q}_7, \hat{q}_{11}\} \subseteq \hat{Q}_{D_E}$. We will compute the maximally permissive data-driven forcing supervisor $V_{K^{\uparrow I}}$ with $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$ based on Algorithm 1. First, by line 3, the uncontrollable event set is updated to $\Sigma_u^+ = \Sigma_u \cup \{\sigma_u^+\}$ as well as the event set $\Sigma^+ = \Sigma \cup \{\sigma_u^+\}$. Then, at line 10, the automaton $\hat{\mathbf{G}}^+$ is built in Fig. 2 based on the data-driven automaton $\hat{\mathbf{G}}(\Sigma, E, D, D^-)$, with states $\hat{q}_1, \hat{q}_7, \hat{q}_{11}$ in $\hat{Q}_{nf} \subseteq \hat{Q}$ (marked with a dashed border) and additional states $q_0^+, q_5^+, q_8^+, q_9^+, q_{10}^+$ (marked in green) from states $\hat{q}_0, \hat{q}_5, \hat{q}_8, \hat{q}_9, \hat{q}_{10}$, respectively. The subautomata $\hat{\mathbf{G}}_{plant}$ and $\hat{\mathbf{G}}_{spec}$ are illustrated in Figs. 3 and 4. From (Reniers and Cai, 2024, Algorithm 1) with $\Sigma_u^+ \subseteq \Sigma^+, \Sigma_{for} \subseteq \Sigma$, $\hat{\mathbf{G}}_{plant}$ as the plant, and $\hat{\mathbf{G}}_{spec}$ as the specification, the maximally permissive, forcibly-controllable supervisor $V_{K^{\uparrow I}}$ can be built, depicted in Fig. 5, where $L(V_{K^{\uparrow I}}) = K^{\uparrow I} = \{\epsilon, f_1, f_1u_2, f_1f_2, f_1f_2f_8\}$.*

By checking informatizability via (Ohtsuka et al., 2026, Algorithm 2), we find that (D, D^-) is not informatizable for E—that is, no non-empty sublanguage $K \subseteq \overline{D_E}$ is controllable wrt. all data-consistent plants. In contrast, our framework yields a non-empty supervisor $V_{K^{\uparrow I}}$ by exploiting the event-forcing mechanism, demonstrating that our approach is generally more permissive than that of Ohtsuka et al. (2026).

Based on the above results, the verification of forcible-informatizability can be straightforwardly derived, as stated in the following corollary.

Corollary 2. *The pair (D, D^-) is forcibly-informatizable for E if and only if $V_{K^{\uparrow I}} \neq \emptyset$ returned by Algorithm 1.*

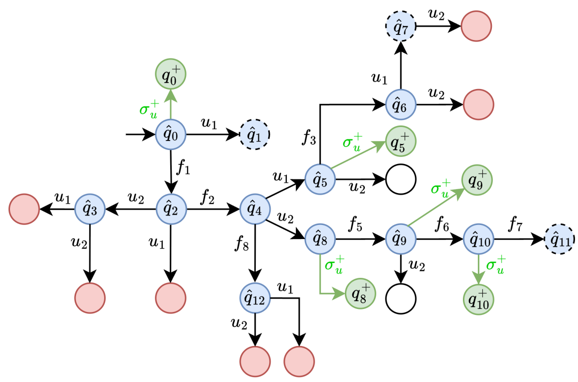

Fig. 2. The automaton $\hat{\mathbf{G}}^+ = (\hat{Q}^+, \Sigma^+, \hat{\delta}^+, \hat{q}_0)$ with states in $\hat{Q}_{D_E}$ in blue, $\hat{Q}_{nf} \subseteq \hat{Q}_{D_E}$ with dashed border, Q_{D^-} in red, and additional states (derived via the virtual uncontrollable event σ_u^+) in green.

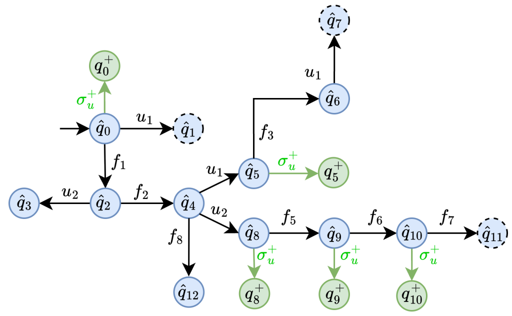

Fig. 3. The subautomaton $\hat{\mathbf{G}}_{plant} \sqsubseteq \hat{\mathbf{G}}^+$ with only states in $\hat{Q}_{D_E}$ and additional states (derived via the virtual uncontrollable event σ_u^+).

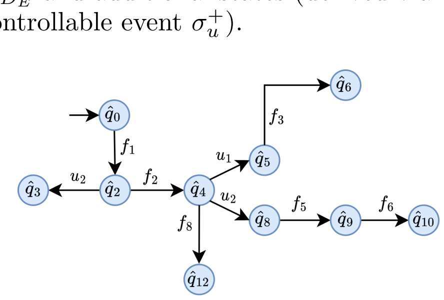

Fig. 4. The subautomaton $\hat{\mathbf{G}}_{spec} \sqsubseteq \hat{\mathbf{G}}_{plant}$, with only states in $\hat{Q}_{D_E} \setminus \hat{Q}_{nf}$.

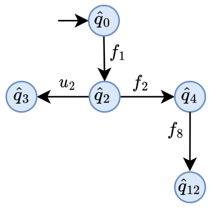

Fig. 5. The maximally permissive, forcibly-controllable supervisor $V_{K^{\uparrow I}}$ where $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$.

Corollary 2 provides a method for determining forcible-informatizability for a given pair (D, D^-) and specification E, alternative to the approach in (Gu et al., 2025, Algorithm 2). For example, based on the results in Example 4, since the maximally permissive data-driven forcing supervisor $V_{K^{\uparrow I}} \neq \emptyset$, the data pair (D, D^-) is forcibly-informatizable for E.

Proof [Proof of Theorem 1]

From Section 2 (results in (Reniers and Cai, 2024)), for the plant $\hat{\mathbf{G}}_{plant}$ and the language $L(\hat{\mathbf{G}}_{spec})$, the set of forcibly-controllable sublanguages can be denoted as

$$\mathcal{F}\big(L(\hat{\mathbf{G}}_{spec})\big) = \left\{ F \subseteq L(\hat{\mathbf{G}}_{spec}) \;\middle|\; \begin{array}{l} F \text{ is forcibly-controllable} \\ \text{with respect to } \hat{\mathbf{G}}_{plant} \end{array} \right\}$$

Further, since forcible-controllability is closed under union (Reniers and Cai, 2024), the set $\mathcal{F}(L(\hat{\mathbf{G}}_{spec}))$ contains a unique supremal element

$$\sup\mathcal{F}(L(\hat{\mathbf{G}}_{spec})) = \bigcup_{F \in \mathcal{F}(L(\hat{\mathbf{G}}_{spec}))} F.$$

Therefore, by line 13 in Algorithm 1, we have $L(V_{K^{\uparrow I}}) = \sup\mathcal{F}(L(\hat{\mathbf{G}}_{spec}))$. Since $K^{\uparrow I}$ is the supremal element of $\mathcal{I}_{for}(D, E)$, to prove $L(V_{K^{\uparrow I}}) = K^{\uparrow I}$, it suffices to prove:

$$\mathcal{F}(L(\hat{\mathbf{G}}_{spec})) = \mathcal{I}_{for}(D, E).$$

($\subseteq$) We first prove $\mathcal{F}(L(\hat{\mathbf{G}}_{spec})) \subseteq \mathcal{I}_{for}(D, E)$. Consider a language $F \in \mathcal{F}(L(\hat{\mathbf{G}}_{spec}))$. Thus, we have $F \subseteq L(\hat{\mathbf{G}}_{spec}) \subseteq L(\hat{\mathbf{G}}_{plant})$. By Definition 1, the following holds:

$$(\forall s \in \overline{F}, \forall \sigma \in \Sigma_u^+)\ s\sigma \in L(\hat{\mathbf{G}}_{plant}) \Rightarrow [s\sigma \in \overline{F}] \vee \\ [((\exists f \in \Sigma_{for}) sf \in \overline{F}) \wedge ((\forall \sigma' \in \Sigma^+ \setminus \Sigma_{for}) s\sigma' \notin \overline{F})]. \quad (7)$$

From line 12, we have $\hat{\mathbf{G}}_{spec} = (\hat{Q}'', \Sigma^+, \hat{\delta}'', \hat{q}_0) \sqsubseteq \hat{\mathbf{G}}_{plant}$ where $\hat{Q}'' = \hat{Q}_{D_E} \setminus \hat{Q}_{nf}$. Thus, $F \subseteq \overline{D_E} \setminus L_{nf}$ holds, where

$$L_{nf} = \{t \in \overline{D_E} \mid (\exists \sigma_u \in \Sigma_u)[t\sigma_u \notin \overline{D_E} \cup D^-] \wedge \\ [(\nexists f' \in \Sigma_{for}) tf' \in \overline{D_E}]\},$$

comprises of strings corresponding to states in the non-forcibly-informative state set $\hat{Q}_{nf}$. To prove $F \in \mathcal{I}_{for}(D, E)$, we need to prove that the data pair (F, D^-) is forcibly-informative for E.

Consider a string $s \in \overline{F}$ and an uncontrollable event $\sigma \in \Sigma_u^+$. First, assume that $s\sigma \in L(\hat{\mathbf{G}}_{plant})$. By (7), the following can be derived:

$$[s\sigma \in \overline{F}] \vee \\ [((\exists f \in \Sigma_{for}) sf \in \overline{F}) \wedge ((\forall \sigma' \in \Sigma^+ \setminus \Sigma_{for}) s\sigma' \notin \overline{F})]. \quad (8)$$

Then, consider $s\sigma \notin L(\hat{\mathbf{G}}_{plant})$. There are three scenarios: (i) $s\sigma \in \overline{D} \setminus \overline{D_E}$; (ii) $\neg\hat{\delta}^+(q_s, \sigma)!$ in $\hat{\mathbf{G}}^+$; (iii) $s\sigma \in D^-$.

For scenario (i), since $\overline{D} \subseteq \Sigma^*$, it follows that $\sigma \in \Sigma_u$. Moreover, as $s\sigma \notin \overline{F}$ and $s\sigma \notin D^-$, informativity fails. According to Algorithm 1 (lines 5–7), there exists $\sigma_u^+ \in \Sigma_u^+$ such that a corresponding state q_s^+ will be added in $\hat{G}^+$ with the transition $(q_s, \sigma_u^+) \rightarrow q_s^+$. Then, by Algorithm 1 (line 11), state $q_s^+ \in \hat{Q}'$ also appears in $\hat{\mathbf{G}}_{plant}$, implying that $s\sigma_u^+ \in L(\hat{\mathbf{G}}_{plant})$; hence (8) follows.

Scenario (ii) is analogous to (i), since it also leads to $s\sigma \notin \overline{F}$ and $s\sigma \notin D^-$. For scenario (iii), $s\sigma \in D^-$ holds directly. Finally, by combining all three scenarios, we obtain

$$[s\sigma \in \overline{F} \cup D^-] \vee [((\exists f \in \Sigma_{for}) sf \in \overline{F}) \wedge \\ ((\forall \sigma' \in \Sigma^+ \setminus \Sigma_{for}) s\sigma' \notin \overline{F})].$$

Given that $\Sigma^+ = \Sigma \cup \{\sigma_u^+\}$ and $s\sigma_u^+ \notin F \subseteq L(\hat{\mathbf{G}}_{spec})$, we deduce

$$[s\sigma \in \overline{F} \cup D^-] \vee [((\exists f \in \Sigma_{for}) sf \in \overline{F}) \wedge \\ ((\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{F})],$$

indicating the forcible-informativity of (F, D^-) for E.

($\supseteq$) We prove that $\mathcal{F}(L(\hat{\mathbf{G}}_{spec})) \supseteq \mathcal{I}_{for}(D, E)$. Consider a sublanguage $K \in \mathcal{I}_{for}(D, E)$ with $K \subseteq \overline{D_E}$. We then prove that $K \in \mathcal{F}(L(\hat{\mathbf{G}}_{spec}))$ where $\mathcal{F}(L(\hat{\mathbf{G}}_{spec})) = \{F \subseteq L(\hat{\mathbf{G}}_{spec}) \mid F$ is forcibly-controllable with respect to $\hat{\mathbf{G}}_{plant}\}$. First, since $\hat{\mathbf{G}}_{plant} = (\hat{Q}', \Sigma^+, \hat{\delta}', \hat{q}_0)$ where $\hat{Q}' = \hat{Q}_{D_E} \cup \{q^+ \in \hat{Q}^+ \mid (\exists q \in \hat{Q}_{D_E}, \exists \sigma_u^+ \in \Sigma_u^+) \hat{\delta}(q, \sigma_u^+) = q^+\}$, $D' \subseteq L(\hat{\mathbf{G}}_{plant})$ holds.

Consider a string $s \in \overline{K}$. Since (K, D^-) is forcibly-informative for K, $s \in \overline{D_E} \setminus L_{nf}$, where

$$L_{nf} = \{t \in \overline{D_E} \mid (\exists \sigma_u \in \Sigma_u)[t\sigma_u \notin \overline{D_E} \cup D^-] \wedge \\ [(\nexists f' \in \Sigma_{for}) tf' \in \overline{D_E}]\}.$$

This implies $K \subseteq L(\hat{\mathbf{G}}_{spec})$. Also, by Proposition 1, for any $\sigma \in \Sigma_u$, the following holds:

$$[s\sigma \in \overline{K} \cup D^-] \vee [((\exists f \in \Sigma_{for}) sf \in \overline{K}) \wedge \\ ((\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K})].$$

Suppose $s\sigma \in L(\hat{\mathbf{G}}_{plant})$, then either $s\sigma \in \overline{K}$ holds or there exists $f \in \Sigma_{for}$ such that

$$[sf \in \overline{K}] \wedge [(\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K})].$$

For the uncontrollable event $\sigma_u^+ \in \Sigma_u^+$, under the condition that $s\sigma_u^+ \in L(\hat{\mathbf{G}}_{plant})$, the following can be indicated based on lines 4–9 in Algorithm 1:

$$(\exists f' \in \Sigma_{for}) sf' \in \overline{D_E}.$$

Since (K, D^-) is forcibly-informative for K and $s\sigma_u^+ \notin \overline{K} \cup D^-$, we conclude

$$[sf' \in \overline{K}] \wedge [(\forall \sigma' \in \Sigma \setminus \Sigma_{for}) s\sigma' \notin \overline{K})].$$

In summary, $K \subseteq L(\hat{\mathbf{G}}_{spec})$ is forcibly-controllable with respect to $\hat{\mathbf{G}}_{plant}$, which concludes the proof.

6. DISCUSSION

We clarify a structural property of the proposed framework in this paper. Since D is finite, the synthesized supervisor $V_{K^{\uparrow I}}$ enforces a finite sublanguage $K^{\uparrow I} \subseteq \overline{D_E}$. This corresponds to *specification termination* rather than state deadlock. That is to say, by construction, at every non-terminal $s \in K^{\uparrow I}$, forcible-informativity of (D, D^-) guarantees an admissible continuation (Proposition 1), and the termination at the boundary of $K^{\uparrow I}$ realizes the specification's finite scope. The finite-horizon nature is a consequence of Assumption 1, which enables decidable verification of forcible-informativity and a supervisor that is maximally permissive wrt. the information contained in D. Extending beyond $\overline{D}$ requires either additional prior knowledge on the true plant $\mathbf{G}$ and/or further data, motivating future work on *dynamic data-driven supervisory control*, where D is updated dynamically and $K^{\uparrow I}$ may also update accordingly.

7. CONCLUSION

In this paper, we explored maximally permissive data-driven supervisory control for DESs using the event-forcing mechanism. Building on the concepts of forcible-informativity and forcible-informatizability, we investigated maximally permissive data-driven forcing supervisory control to enforce a smaller specification with minimal restrictions on permissiveness. The concept of non-forcibly-informative states was introduced, along with an algorithm for its computation. Based on this, we proposed an algorithm to derive the maximally permissive data-driven forcing supervisor by leveraging a model-based forcing supervisory control strategy. Future work includes investigating data-driven supervisory control of DESs under partial observation, with and without forcible events.

REFERENCES

Brandin, B. and Wonham, W.M. (1994). Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic control*, 39(2), 329–342.

Cai, K. (2022). Data-driven supervisory control of discrete-event systems. *Systems, Control and Information*, 66(9), 359–364.

Cassandras, C.G. and Lafortune, S. (2021). *Introduction to Discrete Event Systems*. Springer Nature.

Farooqui, A., Hagebring, F., and Fabian, M. (2021). Mides: A tool for supervisor synthesis via active learning. In *Proceedings of the 17th IEEE International Conference on Automation Science and Engineering*, 792–797. IEEE.

Gu, C., Gao, C., and Cai, K. (2025). Data-driven supervisory control of discrete-event systems with forcible events. `https://www.caikai.org/publication/GuGaoCai25_TAC.pdf`.

Gu, C., Gao, C., and Cai, K. (2024). Data-driven supervisory control of discrete-event systems with forcible events. In *Proceedings of the IFAC Workshop on Discrete Event Systems*, volume 58, 120–125.

Miura, S. and Takai, S. (2018). Decentralized control of timed discrete event systems with conditional decisions for enforcement of forcible events. In *Proceedings of the 57th IEEE Conference on Decisions and Control*, 3956–3961. IEEE.

Ohtsuka, T., Cai, K., and Kashima, K. (2023). Data-informativity for data-driven supervisory control of discrete-event systems. In *Proceedings of the 62nd IEEE Conference on Decisions and Control*, 6917–6922.

Ohtsuka, T., Cai, K., and Kashima, K. (2026). Data-informativity for data-driven supervisory control of discrete-event systems. *IEEE Transactions on Automatic Control*, 71(2), 709–721.

Rashidinejad, A., Reniers, M., and Fabian, M. (2023). Supervisory control synthesis of timed automata using forcible events. *IEEE Transactions on Automatic Control*.

Reniers, M. and Cai, K. (2024). Supervisory control theory with event forcing. *IEEE Transactions on Automatic Control*, 70(5), 3471–3477.

Van Waarde, H.J., Eising, J., Trentelman, H.L., and Camlibel, M.K. (2020). Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11), 4753–4768.

Wardi, Y., Cassandras, C.G., and Cao, X.R. (2018). Perturbation analysis: A framework for data-driven control and optimization of discrete event and hybrid systems. *Annual Reviews in Control*, 45, 267–280.

Wonham, W.M. and Cai, K. (2019). *Supervisory Control of Discrete-Event Systems*. Springer, Cham.