Supervisory Control under Partial Observation: Where Observation Consistency Becomes Decidable

Shaowen Miao, Jan Komenda, Tomáš Masopust, and Yiding Ji

***Abstract*— Observation consistency (OC) and modified observation consistency (MOC) are structural conditions used in hierarchical and modular supervisory control under partial observation. Their verification for languages generated by deterministic finite automata is PSPACE-hard, whereas decidability was open. We answer this question by showing that both problems are undecidable, that is, there are no algorithms verifying OC or MOC. The reductions use only five events, two observable and three high-level, and undecidability persists for every fixed number of shared events. On the positive side, we show that restricting the two alphabets can only produce classes on which both conditions hold trivially, and we identify a decidable class defined by a restriction on the plant: if no cycle of the automaton is labeled exclusively by events outside the shared alphabet, then both conditions are PSPACE-complete.**

***Index Terms*— Discrete-event systems, partial observation, observation consistency, rational relations.**

I. Introduction

Supervisory control of large discrete-event systems (DESs) is often limited by the size of the plant model. Hierarchical control addresses this limitation by synthesizing the supervisor on a smaller high-level abstraction of the plant, and by implementing it on the original, low-level plant. For this to work, properties established at the high level must transfer to the low level.

Under partial observation, a specification admits a supervisor if and only if it is controllable and observable. Observability is not preserved under union, and hence a specification need not have a supremal observable sublanguage. Normality is a stronger condition that is preserved under union and therefore admits a supremal element, which makes it the property of choice in synthesis [1], [2]. Transferring observability or normality between the two levels requires that the abstraction be compatible with the observations available at the low level.

Boutin et al. [3] introduced *observation consistency* (OC) as such a compatibility condition, and showed that it is sufficient for observability to be preserved between the plant and its abstraction. Informally, OC requires that any two high-level strings that the high-level observer cannot distinguish have low-level realizations that the low-level observer cannot distinguish either. Komenda and Masopust [4] studied the same question for normality and observed that OC is too weak for that purpose: normality of the high-level supervisor transfers to the low level only if the low-level realization of one of the two strings may be prescribed in advance. They therefore introduced *modified observation consistency* (MOC), which differs from OC in the order of the quantifiers: for *every* low-level realization of the first string there must be a matching realization of the second one. Under MOC, the computation of supremal normal sublanguages commutes with the high-level projection, which yields maximal permissiveness with respect to the low-level plant.

Both conditions have since been used well beyond their original setting. Miao et al. [5] replaced projections by the set-valued observations that arise from communication delays and losses or from sensor attacks, and introduced the corresponding conditions NOC and MNOC, thereby extending hierarchical synthesis to networked and cyber-attacked DESs. In modular systems, where the monolithic plant may be exponentially larger than each local component, MOC guarantees that the global nonblocking and maximally permissive normal supervisor is the parallel composition of the local normal supervisors [6]. The same condition transfers (L, P)-normality to local projections, which yields a modular computation of supremal (L, P)-normal sublanguages and modular enforcement of critical observability and non-weak opacity [7], as well as of diagnosability and prognosability [8].

All these applications share one obstacle. Since no algorithm verifying MOC was known, they resort to sufficient conditions that are easy to check; namely, alphabetic restrictions [4], [6]. Those conditions are strictly stronger than MOC, and therefore exclude systems that satisfy MOC but violate the alphabet restrictions. What was known about verification is that it is PSPACE-hard [4], which leaves open whether a verification algorithm exists at all.

We show that no such algorithm exists: verification of OC and of MOC is undecidable. The proofs proceed through ratio-

Supported by National Natural Science Foundation of China grants 62303389, 62373289; Guangdong Basic and Applied Basic Research Funding grant 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Youth Science and Technology Talent Support Program of Guangdong Provincial Association for Science and Technology grant SKXRC2025463; Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things grant 2023B1212010007; and by RVO 67985840. *(Corresponding authors: Yiding Ji; Tomáš Masopust)*

S. Miao and Y. Ji are with Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. *smiao585@connect.hkust-gz.edu.cn, jiyiding@hkust-gz.edu.cn*

J. Komenda is with the Institute of Mathematics of the Czech Academy of Sciences, 115 67 Prague, Czechia. *komenda@ipm.cz*

T. Masopust is with the Faculty of Science, Palacky University Olomouc, Czechia. *tomas.masopust@upol.cz*

nal relations, whose universality problem is undecidable [9], [10]. Given an arbitrary rational relation, we construct a DFA such that the generated language is MOC (respectively OC) if and only if the relation is universal. The constructions use five events, two of them observable and three high-level, so that undecidability does not rest on large alphabets.

These results are negative, and they raise the question of what remains verifiable. We show that no answer can be read off the two alphabets: requiring that every high-level string determine its low-level observation forces every observable event of the plant to be a high-level event, and both conditions then hold for a structural reason. A decidable class must therefore restrict the plant. We exhibit one, namely the plants in which the two projections resynchronize after a bounded number of events, and show that both conditions are then PSPACE-complete.

Section II recalls the required notions. Section III characterizes both conditions by sets of low-level observations and by an inclusion of relations. Section IV contains the two reductions and their consequences for the alphabets. Section V delimits what alphabet restrictions can achieve and presents the decidable class.

II. Preliminaries

An alphabet Σ is a finite set of events, and let Σ^* denote the free monoid of all finite strings over Σ, including the empty string ε. A language is a subset of Σ^*. The prefix closure of L is $\overline{L} = \{u \in \Sigma^* \mid uv \in L \text{ for some } v \in \Sigma^*\}$. A projection $\pi\colon \Sigma^* \to \Gamma^*$, where $\Gamma \subseteq \Sigma$, is the morphism for concatenation such that $\pi(a) = a$ for $a \in \Gamma$ and $\pi(a) = \varepsilon$ otherwise. We write Γ^+ for the set of nonempty strings over Γ.

A *(decision) problem* is a language A over Σ, where $x \in A$ means that x is a yes-instance. The problem A is *decidable* if there is an algorithm that halts on every input and accepts exactly the yes-instances; otherwise, A is *undecidable*. A *reduction* from A to B is a computable function f such that $x \in A$ if and only if $f(x) \in B$; if B is decidable and A reduces to B, then A is decidable as well. A *polynomial-space* algorithm is an algorithm whose working space is bounded by a polynomial in the size of the input, and PSPACE is the class of problems admitting such an algorithm. A problem is PSPACE-hard if every problem from PSPACE reduces to it by a polynomial-time reduction. If a PSPACE-hard problem belongs to PSPACE, we talk about a PSPACE-complete problem.

A deterministic finite automaton (DFA) is a tuple $G = (X, \Sigma, \delta, x_0, X_m)$, where X is a finite set of states, Σ is an alphabet, $x_0 \in X$ is the initial state, X_m is the set of marked states, and $\delta\colon X \times \Sigma \to X$ is a partial transition function, extended to strings in the usual way. The generated language of G is $L(G) = \{s \in \Sigma^* \mid \delta(x_0, s) \text{ is defined}\}$, and its marked language is $L_m(G) = \{s \in L(G) \mid \delta(x_0, s) \in X_m\}$. Thus, $L(G)$ is prefix-closed, and $L(G) = L_m(G)$ if every reachable state is marked. A DFA is *trim* if every state is reachable from its initial state and co-reachable to a marked state. A *nondeterministic finite automaton* (NFA) differs from a DFA in that δ maps a state and an event to a set of states; its accepted language is denoted by $\mathcal{L}(\cdot)$.

Let $\Sigma_o, \Sigma_{\text{hi}} \subseteq \Sigma$ be the observable and high-level alphabets. We use the projections $P\colon \Sigma^* \to \Sigma_o^*$ and $Q\colon \Sigma^* \to \Sigma_{\text{hi}}^*$. Set

$$\begin{tikzcd}
\Sigma^* \arrow[r, "Q"] \arrow[d, "P"'] & \Sigma_{\text{hi}}^* \arrow[d, "P_{\text{hi}}"] \\
\Sigma_o^* \arrow[r, "Q_o"] & I^*
\end{tikzcd}$$

Fig. 1. The used projections.

$I = \Sigma_o \cap \Sigma_{\text{hi}}$, and let $Q_o\colon \Sigma_o^* \to I^*$ and $P_{\text{hi}}\colon \Sigma_{\text{hi}}^* \to I^*$ be the corresponding restrictions, see Fig. 1. The projections satisfy

$$Q_o P = P_{\text{hi}} Q. \tag{1}$$

We recall OC and MOC in the form used below [4].

Definition 1: A prefix-closed language $L \subseteq \Sigma^*$ is *observation consistent (OC)* with respect to Q, P, and P_{hi} if, for all $t, t' \in Q(L)$ such that $P_{\text{hi}}(t) = P_{\text{hi}}(t')$, there are $s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$.

Definition 2: A prefix-closed language $L \subseteq \Sigma^*$ is *modified observation consistent (MOC)* with respect to Q, P, and P_{hi} if, for every $s \in L$ and every $t' \in Q(L)$ such that $P_{\text{hi}}(Q(s)) = P_{\text{hi}}(t')$, there is $s' \in L$ such that $Q(s') = t'$ and $P(s') = P(s)$.

The difference is the order of the quantifiers. OC may choose both low-level representatives, whereas MOC must preserve the observation of the given string s.

It is known that MOC implies OC [4, Lemma 10].

A. Rational Relations

Let X and Y be alphabets. A relation $R \subseteq X^* \times Y^*$ is *rational* if there are a finite alphabet C, a regular language $H \subseteq C^*$, and morphisms $f\colon C^* \to X^*$ and $g\colon C^* \to Y^*$ s.t.

$$R = \{(f(k), g(k)) \mid k \in H\}. \tag{2}$$

Rational relations are rational subsets of the monoid $X^* \times Y^*$. Consequently, they are closed under union, concatenation, and Kleene star, and the Cartesian product $K \times M$ of two regular languages $K \subseteq X^*$ and $M \subseteq Y^*$ is rational [11, Chapter III], [12, Chapter IV].

If X and Y are disjoint, every rational relation R admits a representation by a regular language $K_R \subseteq (X \cup Y)^*$ s.t.

$$R = \{(\pi_X(w), \pi_Y(w)) \mid w \in K_R\}, \tag{3}$$

where $\pi_X\colon (X \cup Y)^* \to X^*$ and $\pi_Y\colon (X \cup Y)^* \to Y^*$ are projections. The two representations can be effectively computed from each other [11, Chapter III], [12, Chapter IV].

For $R \subseteq X^* \times Y^*$, let $\operatorname{range}(R) = \{y \in Y^* \mid (x, y) \in R \text{ for some } x \in X^*\}$. The relation is *universal* if $R = X^* \times Y^*$. We write $A \,\dot\cup\, B$ for disjoint union.

Given a rational relation by one of the representations above, it is undecidable whether the relation is universal [9], [10].

Both representations (2) and (3) exist for *every* rational relation. The reductions below may therefore start from an arbitrary instance of the universality problem, and no special form of the given relation or of the morphisms f and g is assumed. Representation (3) additionally requires X and Y to be disjoint, which is no restriction: renaming the alphabets by bijections preserves both rationality and universality, and hence we may always pass to disjoint renamed copies.

III. Characterizations of the Conditions

For every $t \in Q(L)$, define

$$\mathcal{O}_L(t) = \{P(s) \mid s \in L,\ Q(s) = t\} \tag{4}$$

to be the set of low-level observations of strings whose high-level projection is t.

Proposition 1: Let $L \subseteq \Sigma^*$ be prefix-closed, then:

1) L is MOC if and only if, for all $t, t' \in Q(L)$,

$$P_{\mathrm{hi}}(t) = P_{\mathrm{hi}}(t') \implies \mathcal{O}_L(t) = \mathcal{O}_L(t')\,. \tag{5}$$

2) L is OC if and only if, for all $t, t' \in Q(L)$,

$$P_{\mathrm{hi}}(t) = P_{\mathrm{hi}}(t') \implies \mathcal{O}_L(t) \cap \mathcal{O}_L(t') \neq \emptyset. \tag{6}$$

Proof: Assume that L is MOC, and let $t, t' \in Q(L)$ have the same P_{hi}-projection. If $u \in \mathcal{O}_L(t)$, there is $s \in L$ such that $Q(s) = t$ and $P(s) = u$, and MOC gives $s' \in L$ with $Q(s') = t'$ and $P(s') = u$, that is, $\mathcal{O}_L(t) \subseteq \mathcal{O}_L(t')$. Interchanging t and t' gives $\mathcal{O}_L(t) = \mathcal{O}_L(t')$. Conversely, assume (5), and let $s \in L$ and $t' \in Q(L)$ satisfy $P_{\mathrm{hi}}(Q(s)) = P_{\mathrm{hi}}(t')$. Then, $P(s) \in \mathcal{O}_L(Q(s)) = \mathcal{O}_L(t')$ and the definition of $\mathcal{O}_L(t')$ gives $s' \in L$ with $Q(s') = t'$ and $P(s') = P(s)$. Thus, L is MOC.

If L is OC, then for every $t, t' \in Q(L)$ with $P_{\mathrm{hi}}(t) = P_{\mathrm{hi}}(t')$, there are $s, s' \in L$ with $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$, and hence $P(s) \in \mathcal{O}_L(t) \cap \mathcal{O}_L(t')$. Conversely, assume (6). If $u \in \mathcal{O}_L(t) \cap \mathcal{O}_L(t')$, there are $s, s' \in L$ with $Q(s) = t$, $Q(s') = t'$, and $P(s) = u = P(s')$, and therefore L is OC. ∎

Remark 1: If $\Sigma_o \cap \Sigma_{\mathrm{hi}} = \emptyset$, then $P_{\mathrm{hi}}(t) = P_{\mathrm{hi}}(t')$ for all $t, t' \in Q(L)$, and the conditions of Proposition 1 read (i) L is MOC if and only if $\mathcal{O}_L(t) = \mathcal{O}_L(t')$ for all $t, t' \in Q(L)$, and (ii) L is OC if and only if $\mathcal{O}_L(t) \cap \mathcal{O}_L(t') \neq \emptyset$ for all $t, t' \in Q(L)$.

For a language $L \subseteq \Sigma^*$ and projections P and Q, we define the relation

$$\mathcal{R}_L = \{(P(s), Q(s)) \mid s \in L\} \tag{7}$$

and the set of compatible pairs as

$$\mathcal{C}_L = \{(u, v) \in P(L) \times Q(L) \mid Q_o(u) = P_{\mathrm{hi}}(v)\}.$$

If L is regular, $\mathcal{R}_L$ is rational, since (7) is of the form (2) with $H = L$, $f = P$, and $g = Q$. By (1), $\mathcal{R}_L \subseteq \mathcal{C}_L$, and we show that the other inclusion characterizes the MOC condition.

Proposition 2: A prefix-closed language L is MOC if and only if $\mathcal{C}_L \subseteq \mathcal{R}_L$.

Proof: Assume that L is MOC, and let $(u, v) \in \mathcal{C}_L$. Choose $s \in L$ with $P(s) = u$. Then $P_{\mathrm{hi}}(Q(s)) = Q_o(P(s)) = Q_o(u) = P_{\mathrm{hi}}(v)$. By MOC, there is $s' \in L$ with $P(s') = u$ and $Q(s') = v$, that is, $(u, v) \in \mathcal{R}_L$, and hence $\mathcal{C}_L \subseteq \mathcal{R}_L$.

Conversely, let $s \in L$ and $t' \in Q(L)$ with $P_{\mathrm{hi}}(Q(s)) = P_{\mathrm{hi}}(t')$. Since $P_{\mathrm{hi}}(Q(s)) = Q_o(P(s)) = P_{\mathrm{hi}}(t')$, $(P(s), t') \in \mathcal{C}_L \subseteq \mathcal{R}_L$, and therefore there is $s' \in L$ such that $P(s') = P(s)$ and $Q(s') = t'$, that is, L is MOC. ∎

IV. Main Results

To prove the results, we state two elementary observations.

Lemma 1: Let $K \subseteq V^*$ be a regular language, and let $a \notin V$. Then $L = V^* \cup Ka$ is prefix-closed and regular. ∎

Indeed, every proper prefix of a string of Ka belongs to V^*, and every prefix of a string of V^* belongs to V^*; regularity is clear. A DFA for L can be constructed from the DFA for K in polynomial time. Indeed, we first complete the DFA for K by adding a new sink state, and then add a new state x together with a-transitions to it from every marked state.

The next construction combines standard closure properties of rational relations with a block encoding. It reduces every instance of the universality problem to an instance over two-letter alphabets, which is what makes the alphabets of the reductions below fixed and small.

Lemma 2: Let $R \subseteq X^* \times Y^*$ be rational, and let $A = \{a_1, a_2\}$ and $B = \{b_1, b_2\}$ be two-letter alphabets. There is an effectively constructible rational relation $S \subseteq A^* \times B^*$ such that $\mathrm{range}(S) = B^*$ and $S = A^* \times B^* \iff R = X^* \times Y^*$.

Proof: Choose $m \geq 2$ such that $2^m \geq \max\{|X|, |Y|\}$, and let $h\colon X^* \to A^*$ and $h'\colon Y^* \to B^*$ be morphisms that map every event to a string of length m and are injective on events. Then h and h' are injective, and $h(X^*)$ and $h'(Y^*)$ are regular. Let

$$D_A = a_1 h(X^*), \qquad D_B = b_1 h'(Y^*),$$

which are regular as well, and define

$$\begin{aligned} S = {} & \{(a_1 h(x), b_1 h'(y)) \in D_A \times D_B \mid (x, y) \in R\} \\ & \cup\ (A^* \setminus D_A) \times B^* \\ & \cup\ A^* \times (B^* \setminus D_B)\,. \end{aligned}$$

The relation $\{(h(x), h'(y)) \mid (x, y) \in R\}$ is rational, because writing R in the form (2) exhibits it in the form (2) with the morphisms $h \circ f$ and $h' \circ g$. The first term of S is its concatenation with a finite relation; in the other two terms, both factors are regular languages, and their Cartesian product is rational. Since rational relations are closed under concatenation and union, S is rational [11, Chapter III], [12, Chapter IV].

Every string of D_A is nonempty, and therefore $\varepsilon \notin D_A$. The second term then gives $(\varepsilon, v) \in S$ for every $v \in B^*$, and hence $\mathrm{range}(S) = B^*$.

Finally, the last two terms of S contain every pair outside $D_A \times D_B$. Every string of D_A is $a_1 h(x)$ for a unique $x \in X^*$, because h is injective, and similarly every string of D_B is $b_1 h'(y)$ for a unique $y \in Y^*$. Thus, for pairs in $D_A \times D_B$,

$$(a_1 h(x), b_1 h'(y)) \in S \iff (x, y) \in R\,,$$

and therefore S is universal if and only if R is universal. ∎

The lemma does not require A and B to be disjoint. The proof of Theorem 1 uses disjoint A and B, since it applies (3), whereas the reduction of Section IV-B uses $A = B$.

A. Undecidability of MOC Verification

We now reduce universality of rational relations to MOC verification using the two preceding constructions.

Theorem 1: Given a DFA G and event sets $\Sigma_o, \Sigma_{\rm hi} \subseteq \Sigma$, it is undecidable whether $L(G)$ is MOC. The result holds even if $|\Sigma| = 5$, $|\Sigma_o| = 2$, and $|\Sigma_{\rm hi}| = 3$.

Proof: Let $R \subseteq X^* \times Y^*$ be a rational relation, let $A = \{0,1\}$ and $B = \{2,3\}$, and let $S \subseteq A^* \times B^*$ be the relation provided by Lemma 2. Since A and B are disjoint, (3) applies to S: writing $U = A \,\dot\cup\, B$, there is a regular language $K \subseteq U^*$ such that

$$S = \{(\pi_A(w), \pi_B(w)) \mid w \in K\}.$$

Take a fresh event a, and define

$$\Sigma = U \,\dot\cup\, \{a\}, \qquad \Sigma_o = A, \qquad \Sigma_{\rm hi} = B \,\dot\cup\, \{a\}.$$

Then $\Sigma = \Sigma_o \,\dot\cup\, \Sigma_{\rm hi}$ and $\Sigma_o \cap \Sigma_{\rm hi} = \emptyset$. Let

$$L = U^* \cup Ka\,. \tag{8}$$

By Lemma 1, L is prefix-closed and effectively represented by a DFA.

Since $a \in \Sigma_{\rm hi}$, the high-level projections of strings of Ka are the strings $\pi_B(k)a$ with $k \in K$, that is, the strings ta with $t \in \mathrm{range}(S) = B^*$. Together with $Q(U^*) = B^*$, this gives

$$Q(L) = B^* \cup B^* a.$$

Fix $t \in B^*$. For every $u \in A^*$, the string ut belongs to U^* and satisfies $P(ut) = u$ and $Q(ut) = t$; conversely, $P(U^*) \subseteq A^*$. The high-level projection of every string of Ka ends with a, and hence differs from t, so the strings of Ka do not contribute to $\mathcal{O}_L(t)$. Therefore,

$$\mathcal{O}_L(t) = A^*. \tag{9}$$

A string of L with high-level projection ta is of the form ka with $k \in K$ and $\pi_B(k) = t$, and, as $a \notin \Sigma_o$, its observation is $\pi_A(k)$. Hence

$$\mathcal{O}_L(ta) = \{x \in A^* \mid (x,t) \in S\}. \tag{10}$$

Since $\Sigma_o \cap \Sigma_{\rm hi} = \emptyset$, Remark 1 says that L is MOC if and only if all the sets (9) and (10) coincide, that is, if and only if $\{x \mid (x,t) \in S\} = A^*$ for every $t \in B^*$. The latter says that $S = A^* \times B^*$. By Lemma 2,

$$L \text{ is MOC} \iff S = A^* \times B^* \iff R = X^* \times Y^*.$$

Hence an algorithm deciding MOC would decide universality of rational relations, which is a contradiction. ∎

B. Undecidability of OC Verification

Let $R_0 \subseteq X_0^* \times Y_0^*$ be rational, and let $V = \{2,3\}$. Applying Lemma 2 with $A = B = V$ yields a rational relation $R \subseteq V^* \times V^*$ that is universal if and only if R_0 is universal. Write R in the form (2), that is,

$$R = \{(f(k), g(k)) \mid k \in H\}, \qquad H \subseteq C^* \text{ regular},$$

and let $e\colon C^* \to \{0,1\}^*$ be a morphism that maps every event of C to a string of length $m \geq 2$ and is injective on events; then e is injective, and the length of every string of $e(C^*)$ is divisible by m. Put

$$Z = \{0,1\}, \qquad W = V = \{2,3\},$$

and let the four auxiliary events of the construction be

$$q = 0, \qquad d = 1, \qquad \alpha = 2, \qquad \beta = 3\,.$$

Define the morphisms

$$h_f(c) = e(c)f(c), \qquad h_g(c) = e(c)g(c), \qquad c \in C,$$

with codomain $(Z \cup W)^*$. The language

$$J = q\alpha V^* \;\cup\; d\beta V^* \;\cup\; \alpha h_f(H) \;\cup\; \beta h_g(H) \tag{11}$$

is regular. Let $\pi_Z\colon (Z \cup W)^* \to Z^*$ and $\pi_W\colon (Z \cup W)^* \to W^*$ denote the projections. Since $e(c) \in Z^*$ and $f(c), g(c) \in W^*$, we have, for every $k \in H$,

$$\begin{aligned} \pi_Z(h_f(k)) &= e(k), & \pi_W(h_f(k)) &= f(k), \\ \pi_Z(h_g(k)) &= e(k), & \pi_W(h_g(k)) &= g(k). \end{aligned}$$

The four languages in (11) are pairwise disjoint, because their strings begin with 0, 1, 2, and 3, respectively. Consequently, the relation induced by the projections on J consists of the following pairs:

$$\begin{array}{llll} (q, \alpha x) & x \in V^*, & (d, \beta y) & y \in V^*, \\ (e(k), \alpha f(k)) & k \in H, & (e(k), \beta g(k)) & k \in H. \end{array} \tag{12}$$

Consequently, the associated observation sets are

$$\mathcal{O}_J(\alpha x) = \{q\} \cup e(\{k \in H \mid f(k) = x\}), \tag{13}$$

$$\mathcal{O}_J(\beta y) = \{d\} \cup e(\{k \in H \mid g(k) = y\}). \tag{14}$$

Here $\mathcal{O}_J$ is defined as in (4), using the projections from $(Z \cup W)^*$ to Z^* and W^*.

The word q belongs to $\mathcal{O}_J(\alpha x)$ for every $x \in V^*$, and d belongs to $\mathcal{O}_J(\beta y)$ for every $y \in V^*$. The one-letter words q and d are distinct, and neither belongs to $e(C^*)$, whose strings have length divisible by $m \geq 2$. These two observation sets therefore intersect if and only if they contain a common string $e(k)$; since e is injective, it is if and only if a single $k \in H$ satisfies $f(k) = x$ and $g(k) = y$. Hence,

$$\mathcal{O}_J(\alpha x) \cap \mathcal{O}_J(\beta y) \neq \emptyset \iff (x, y) \in R\,. \tag{15}$$

We are now ready to prove the main result for OC.

Theorem 2: Given a DFA G and event sets $\Sigma_o, \Sigma_{\rm hi} \subseteq \Sigma$, it is undecidable whether $L(G)$ is OC. The result holds even if $|\Sigma| = 5$, $|\Sigma_o| = 2$, and $|\Sigma_{\rm hi}| = 3$.

Proof: Introduce a fresh event a and set

$$\Sigma = Z \dot\cup W \dot\cup \{a\} = \{0,1,2,3,a\}, \;\; \Sigma_o = Z, \;\; \Sigma_{\rm hi} = W \dot\cup \{a\}.$$

Then $\Sigma = \Sigma_o \,\dot\cup\, \Sigma_{\rm hi}$ and $I = \Sigma_o \cap \Sigma_{\rm hi} = \emptyset$. Define

$$L = (Z \cup W)^* \cup Ja. \tag{16}$$

Lemma 1 shows that L is prefix-closed, regular, and effectively represented by a DFA. Since $a \in \Sigma_{\rm hi}$,

$$Q(L) = W^* \cup (\alpha V^* \cup \beta V^*)a = W^* \cup W^+ a,$$

where the last equality holds because $\alpha = 2$ and $\beta = 3$ are the two events of W, thus each string of W^+ lies in exactly one of the two families according to its first event.

For every $t \in W^*$, only the language $(Z \cup W)^*$ contributes, because the high-level projection of every string of Ja ends with a. As $(Z \cup W)^*$ realizes every low-level observation together with every high-level string,

$$\mathcal{O}_L(t) = Z^*.$$

The remaining high-level strings end with a and hence arise only from Ja. Since $a \notin \Sigma_o$, the projection P erases a, and (13) and (14) yield

$$\mathcal{O}_L(\alpha x a) = \{q\} \cup e(\{k \in H \mid f(k) = x\}),$$
$$\mathcal{O}_L(\beta y a) = \{d\} \cup e(\{k \in H \mid g(k) = y\}).$$

All these sets are nonempty subsets of Z^*. Consequently, each of them meets $\mathcal{O}_L(t) = Z^*$; any two sets of the first kind contain q; and any two sets of the second kind contain d. The only pairs left to consider are $\alpha x a$ and $\beta y a$, and by (15), their observation sets intersect if and only if $(x, y) \in R$. Since $\alpha x a \in Q(L)$ and $\beta y a \in Q(L)$ for all $x, y \in V^*$, and since $I = \emptyset$, Remark 1 gives

$$L \text{ is OC} \iff (x, y) \in R \text{ for all } x, y \in V^*$$
$$\iff R = V^* \times V^* \iff R_0 = X_0^* \times Y_0^*,$$

the last equivalence by Lemma 2. An algorithm deciding OC for DFAs would consequently decide universality of rational relations, which is impossible. ■

C. Consequences

Both reductions use the same five events and produce instances with $\Sigma_o \cap \Sigma_{\text{hi}} = \emptyset$. Neither the number of events nor the size of the shared alphabet can therefore be held responsible for the undecidability.

Corollary 1: For every fixed $k \geq 0$, deciding whether $L(G)$ is OC, and deciding whether it is MOC, is undecidable already on instances over a fixed alphabet with $5 + k$ events, of which $2+k$ are observable and $3+k$ are high-level, so that $|\Sigma_o \cap \Sigma_{\text{hi}}| = k$. For $k = 0$, one may take $\Sigma = \{0, 1, 2, 3, a\}$, $\Sigma_o = \{0, 1\}$, and $\Sigma_{\text{hi}} = \{2, 3, a\}$.

Proof: The case $k = 0$ is the second claim of Theorems 1 and 2. For $k \geq 1$, add k fresh events to Σ, to Σ_o, and to Σ_{hi}, without adding any transition to G. The added events occur in no string of $L(G)$, and hence P, Q, Q_o, and P_{hi} act on $L(G)$ exactly as before, so that OC and MOC are unaffected. ■

The alphabets cannot be reduced further along these lines. Both reductions start from universality of rational relations, and a rational relation $R \subseteq \{x\}^* \times \{y\}^*$ over two one-letter alphabets is a rational subset of the commutative monoid $\mathbb{N} \times \mathbb{N}$, and hence semilinear, so that its universality is decidable. At least one of the two components must therefore carry two events. Whether five events in total are optimal is open.

Although neither condition can be verified, a single candidate counterexample can always be checked.

Lemma 3: Let $L = L(G)$ for a DFA G.

1) For fixed $u \in \Sigma_o^*$ and $v \in \Sigma_{\text{hi}}^*$, it is decidable whether

$$u \in P(L),\ v \in Q(L),\ Q_o(u) = P_{\text{hi}}(v),\ \text{and}$$
$$L \cap P^{-1}(\{u\}) \cap Q^{-1}(\{v\}) = \emptyset\,. \tag{17}$$

2) For fixed $t, t' \in Q(L)$, it is decidable whether

$$\mathcal{O}_L(t) \cap \mathcal{O}_L(t') = \emptyset. \tag{18}$$

Moreover, L is not MOC if and only if some pair (u, v) satisfies (17), and L is not OC if and only if some pair $t, t' \in Q(L)$ with $P_{\text{hi}}(t) = P_{\text{hi}}(t')$ satisfies (18).

Proof: The languages $P(L)$, $Q(L)$, $P^{-1}(\{u\})$, and $Q^{-1}(\{v\})$ are regular and can be effectively constructed; hence every condition in (17) is decidable by standard operations on regular languages. Likewise, for a fixed t, the language

$$\mathcal{O}_L(t) = P\big(L \cap Q^{-1}(\{t\})\big)$$

is regular and effectively constructible, so (18) is decidable. The two characterizations of failure restate Proposition 2 and Proposition 1, respectively. ■

A violation can therefore always be found by an exhaustive search through the pairs, ordered by length. What the undecidability results exclude is any computable bound on how far this search must go: if a computable function bounded the length of a shortest violating pair in terms of the length of the encoding of $(G, \Sigma_o, \Sigma_{\text{hi}})$, then inspecting the finitely many pairs up to that length by Lemma 3 would decide both conditions.

V. A Decidable Class

The undecidability proofs exploit that $\mathcal{R}_L$ is an arbitrary rational relation: the two projections $P(s)$ and $Q(s)$ may drift apart without bound. We first observe that this cannot be repaired by restricting the alphabets, and then exhibit a restriction on the language that makes both conditions decidable.

A. Alphabet Restrictions are Degenerate

A candidate for a tractable case is the requirement that every high-level string determine its low-level observation, that is, that $\mathcal{O}_L(t)$ be a singleton for every $t \in Q(L)$. For prefix-closed languages, this is a purely structural restriction, under which both conditions hold automatically.

Proposition 3: Let $L \subseteq \Sigma^*$ be prefix-closed. The following are equivalent:

1) $\mathcal{O}_L(t)$ is a singleton for every $t \in Q(L)$;
2) no event of $\Sigma_o \setminus \Sigma_{\text{hi}}$ occurs in a string of L;
3) $P(s) = P_{\text{hi}}(Q(s))$ for every $s \in L$.

If they hold, then L is MOC, and hence OC.

Proof: (1) ⇒ (2). Assume that $\sigma \in \Sigma_o \setminus \Sigma_{\text{hi}}$ occurs in a string of L, say $u\sigma v \in L$. Since L is prefix-closed, both u and $u\sigma$ belong to L. As $\sigma \notin \Sigma_{\text{hi}}$, $Q(u\sigma) = Q(u)$, whereas $\sigma \in \Sigma_o$ gives $P(u\sigma) = P(u)\sigma \neq P(u)$. Hence $\{P(u), P(u)\sigma\} \subseteq \mathcal{O}_L(Q(u))$, which is therefore not a singleton.

(2) ⇒ (3). Let $s \in L$. By (2), every event of s that belongs to Σ_o belongs to Σ_{hi}, hence to I. Writing π_I for the projection of Σ^* onto I^*, we get $P(s) = \pi_I(s) = P_{\text{hi}}(Q(s))$.

(3) ⇒ (1). By (3), $\mathcal{O}_L(t) = \{P_{\text{hi}}(t)\}$ for every $t \in Q(L)$.

Finally, assume (3) and let $t, t' \in Q(L)$ satisfy $P_{\text{hi}}(t) = P_{\text{hi}}(t')$. Then $\mathcal{O}_L(t) = \{P_{\text{hi}}(t)\} = \{P_{\text{hi}}(t')\} = \mathcal{O}_L(t')$, and Proposition 1 gives MOC. ■

Condition (2) is the containment $\Sigma_o \subseteq \Sigma_{\text{hi}}$, relativized to the events that occur, and the symmetric containment $\Sigma_{\text{hi}} \subseteq \Sigma_o$ forces MOC as well [4]. Restrictions imposed by the two alphabets thus yield only classes for which the answer is affirmative. A decidable class with a nonconstant answer must restrict the language.

B. Bounded Desynchronization

The obstruction identified above is that $P(s)$ and $Q(s)$ may be synchronized arbitrarily loosely. The events of $I = \Sigma_0 \cap \Sigma_{\text{hi}}$ are the only events both projections retain, and they are therefore the only synchronization points available.

Definition 3: A language $L \subseteq \Sigma^*$ is *I-bounded* if there is $N \in \mathbb{N}$ such that no string of L contains more than N consecutive events of $\Sigma \setminus I$.

Lemma 4: Let G be a trim DFA. Then $L(G)$ is I-bounded if and only if G has no cycle all of whose transitions are labeled by events of $\Sigma \setminus I$.

Proof: Indeed, the condition says that the subgraph of G obtained by deleting the transitions labeled by I is acyclic. ■

The condition is decidable in time linear in the size of G. Fix N to be the number of states of G minus one and put

$$B_o = (\Sigma_o \setminus I)^{\leq N}, \qquad B_{\text{hi}} = (\Sigma_{\text{hi}} \setminus I)^{\leq N}.$$

Every $u \in \Sigma_o^*$ factors as $u = p_0 i_1 p_1 \cdots i_n p_n$ with $i_j \in I$ and $p_j \in (\Sigma_o \setminus I)^*$, and similarly every $v \in \Sigma_{\text{hi}}^*$ factors as $v = q_0 i'_1 q_1 \cdots i'_m q_m$. If $Q_o(u) = P_{\text{hi}}(v)$, then $n = m$ and $i_j = i'_j$ for all j, and the two factorizations can be interleaved. Over the finite alphabet

$$\Gamma = (B_o \times B_{\text{hi}}) \,\dot\cup\, I,$$

define the encoding of such a pair by

$$\operatorname{enc}(u, v) = (p_0, q_0)\, i_1\, (p_1, q_1)\, i_2 \cdots i_n\, (p_n, q_n).$$

The map enc is injective on the pairs it is defined for, and it is defined on all of $\mathcal{C}_L$ whenever L is I-bounded: for $s \in L$ the block of s between two consecutive events of I has at most N events, and hence so do the corresponding blocks of $P(s)$ and of $Q(s)$. We write $\operatorname{enc}_o(u) = p_0\, i_1\, p_1 \cdots i_n\, p_n$ and $\operatorname{enc}_{\text{hi}}(v) = q_0\, i_1\, q_1 \cdots i_n\, q_n$ for the corresponding encodings of a single string, over the alphabets $B_o \,\dot\cup\, I$ and $B_{\text{hi}} \,\dot\cup\, I$.

Theorem 3: Let G be a DFA such that $L = L(G)$ is I-bounded. Then it is decidable whether L is OC, and whether L is MOC. Both problems are in PSPACE.

Proof: We show that $\operatorname{enc}(\mathcal{R}_L)$ and $\operatorname{enc}(\mathcal{C}_L)$ are regular and effectively constructible, so that

$$L \text{ is MOC} \iff \operatorname{enc}(\mathcal{C}_L) \subseteq \operatorname{enc}(\mathcal{R}_L)$$

by Proposition 2 and injectivity of enc.

For $\operatorname{enc}(\mathcal{R}_L)$, build the NFA $A_\mathcal{R}$ over Γ with the state set and the initial state of G, with all states accepting, and with the transitions

- $x \xrightarrow{(p,q)} x'$ whenever $\delta(x, w) = x'$ for a $w \in (\Sigma \setminus I)^{\leq N}$ with $P(w) = p$ and $Q(w) = q$;
- $x \xrightarrow{i} x'$ whenever $\delta(x, i) = x'$, for $i \in I$.

Since L is I-bounded and prefix-closed, the accepting runs of $A_\mathcal{R}$ correspond exactly to the I-factorizations of the strings of L, and the label of a run is $\operatorname{enc}(P(s), Q(s))$. Hence $A_\mathcal{R}$ accepts $\operatorname{enc}(\mathcal{R}_L)$.

For $\operatorname{enc}(\mathcal{C}_L)$, the same construction applied to P alone, respectively Q alone, gives NFAs over $B_o \dot\cup I$ and $B_{\text{hi}} \dot\cup I$ accepting the block encodings of $P(L)$ and of $Q(L)$. Let $\rho_o, \rho_{\text{hi}} \colon \Gamma^* \to (B_o \dot\cup I)^*, (B_{\text{hi}} \dot\cup I)^*$ be the morphisms erasing the second, respectively the first, component of the pairs and fixing I. Then

$$\operatorname{enc}(\mathcal{C}_L) = \rho_o^{-1}(\operatorname{enc}_o(P(L))) \cap \rho_{\text{hi}}^{-1}(\operatorname{enc}_{\text{hi}}(Q(L))),$$

because a pair lies in $\mathcal{C}_L$ exactly when its components lie in $P(L)$ and $Q(L)$ and their I-projections agree, the latter being enforced by the common I-letters of the encoding. Regular languages being closed under inverse morphisms and intersection, $\operatorname{enc}(\mathcal{C}_L)$ is regular.

For OC, let $\Gamma_3 = (B_o \times B_{\text{hi}} \times B_{\text{hi}}) \,\dot\cup\, I$ and let A_3 be the product of two copies of $A_\mathcal{R}$ that synchronizes on the first component of the pairs and on the letters of I. Then A_3 accepts the encodings of the triples (u, v, v') such that (u, v) and (u, v') both lie in $\mathcal{R}_L$, that is, such that $u \in \mathcal{O}_L(v) \cap \mathcal{O}_L(v')$. Let π erase the first component. By Proposition 1,

$$L \text{ is OC} \iff D \subseteq \pi(\mathcal{L}(A_3)),$$

where D is the set of encodings of the pairs $(v, v') \in Q(L) \times Q(L)$ with $P_{\text{hi}}(v) = P_{\text{hi}}(v')$, which is regular by the argument used for $\operatorname{enc}(\mathcal{C}_L)$. As π is a letter-to-letter morphism, $\pi(\mathcal{L}(A_3))$ is regular, and the inclusion is decidable.

For the complexity bound, N is bounded by the number of states, n, and hence every letter of Γ is written in polynomial space, and whether $x \xrightarrow{(p,q)} x'$ is a transition of $A_\mathcal{R}$ is decided in polynomial space by guessing w event by event. Both criteria are inclusions between languages of NFAs with n, respectively n^2, states, which is in PSPACE [13], [14]. ■

The class is nontrivial in both directions. Let $\Sigma_o = \{a, c\}$ and $\Sigma_{\text{hi}} = \{x, c\}$, so that $I = \{c\}$. The prefix closure L of $(ac + xc)^*$ is I-bounded with $N = 1$, and it is neither OC nor MOC: the high-level strings c and xc satisfy $P_{\text{hi}}(c) = c = P_{\text{hi}}(xc)$, while $\mathcal{O}_L(c) = \{ac, aca\}$ and $\mathcal{O}_L(xc) = \{c, ca\}$ are disjoint, since every string of the former begins with a and every string of the latter with c. The language c^* is I-bounded and MOC. Every finite language is I-bounded. In contrast, the languages built in Theorems 1 and 2 contain U^*, respectively $(Z \cup W)^*$, with $I = \emptyset$, and are not I-bounded.

Theorem 3 is the first decidable case with a nonconstant answer, and the bound it gives is tight.

Theorem 4: Verification of OC, and verification of MOC, is PSPACE-hard on the I-bounded instances, even if $N = 3$.

Proof: We reduce from the problem whether, given DFAs $A_1, \ldots, A_n$ over an alphabet Δ, $\bigcup_{i=1}^n L_m(A_i) = \Delta^*$, which is PSPACE-complete [15].

For MOC, take fresh pairwise distinct events o, h, and $\tau_1, \ldots, \tau_n$, and set

$$\Sigma = \Delta \dot\cup \{o, h\} \dot\cup \{\tau_1, \ldots, \tau_n\},\ \Sigma_o = \Delta \dot\cup \{o\},\ \Sigma_{\text{hi}} = \Delta \dot\cup \{h\},$$

so that $I = \Delta$ and the events τ_i lie in neither Σ_o nor Σ_{hi} and are therefore erased by both projections. Let

$$L = \Delta^* \cup \Delta^* o \cup \Delta^* h \cup \bigcup_{i=1}^n \tau_i \big(\Delta^* \cup L_m(A_i) o \cup L_m(A_i) oh\big).$$

This language is prefix-closed, and a DFA for it is obtained from a state m carrying a self-loop on every event of Δ, an o-transition and an h-transition from m to two states without successors, a τ_i-transition from m to the initial state of a copy of A_i, an o-transition from every marked state of that copy to a state r_i, and an h-transition from r_i to a state without successors. Note that every A_i is complete, that is, $L(A_i) = \Delta^*$, and hence $\tau_i \Delta^*$ is obtained for free. The DFA has $\sum_i |A_i| + O(n)$ states and is computed in polynomial time.

Since $o \notin \Sigma_{\text{hi}}$, $h \notin \Sigma_o$, and the τ_i are erased, we obtain $P(L) = \Delta^* \cup \Delta^* o$ and $Q(L) = \Delta^* \cup \Delta^* h$, and Q_o and P_{hi} erase o and h, respectively. Hence

$$\mathcal{C}_L = \{(w, w), (w, wh), (wo, w), (wo, wh) \mid w \in \Delta^*\}.$$

The strings w, wo, and wh of L realize the first three families of pairs. A string of L whose observation contains o and whose high-level projection contains h must be of the form $\tau_i w o h$ with $w \in L_m(A_i)$, and it realizes (wo, wh). By Proposition 2,

$$L \text{ is MOC} \iff \bigcup_i L_m(A_i) = \Delta^*.$$

Finally, the events outside $I = \Delta$ are o, h, and the τ_i, and the longest run of them in a string of L is $\tau_i o h$, so L is I-bounded with $N = 3$.

For OC, take fresh pairwise distinct events $o_1, o_2, c, h_1, h_2, \tau_1, \ldots, \tau_n$, and set $\Sigma_o = \Delta \mathbin{\dot\cup} \{o_1, o_2, c\}$ and $\Sigma_{\text{hi}} = \Delta \mathbin{\dot\cup} \{h_1, h_2\}$, again with $I = \Delta$ and with the τ_i erased by both projections. Let L be the prefix closure of

$$\Delta^* o_1 h_1 \cup \Delta^* o_2 h_2 \cup \bigcup_{i=1}^{n} \tau_i L_m(A_i) c \{h_1, h_2\},$$

which is again realized by a DFA of polynomial size. For every $w \in \Delta^*$, the high-level strings wh_1 and wh_2 belong to $Q(L)$ and satisfy $P_{\text{hi}}(wh_1) = w = P_{\text{hi}}(wh_2)$, and

$$\mathcal{O}_L(wh_1) = \{wo_1\} \cup \{wc \mid w \in \textstyle\bigcup_i L_m(A_i)\},$$
$$\mathcal{O}_L(wh_2) = \{wo_2\} \cup \{wc \mid w \in \textstyle\bigcup_i L_m(A_i)\}.$$

As $o_1 \neq o_2$, these two sets intersect if and only if $w \in \bigcup_i L_m(A_i)$. The only other pairs of $Q(L)$ with equal P_{hi}-projection involve a string of Δ^*, and $\mathcal{O}_L(w)$ contains wo_1 and wo_2, so those pairs impose no condition. By Proposition 1, L is OC if and only if $\bigcup_i L_m(A_i) = \Delta^*$. The longest run of events outside Δ is $\tau_i c h_1$, and hence $N = 3$ again. ■

Consequently, both problems are PSPACE-complete on this class. It is open whether I-boundedness can be relaxed to the weaker requirement that $\mathcal{R}_L$ be a synchronous relation, of which it is a sufficient structural condition.

VI. Conclusion

Verification of OC and MOC is undecidable for prefix-closed regular languages represented by DFAs. Both reductions use five events in total, and undecidability persists for every fixed number of high-level and observable events. A single candidate counterexample can always be checked, but no computable function of the input bounds how far the search for one must go. On the positive side, requiring every high-level string to determine its low-level observation forces both conditions to hold, so that no restriction of the two alphabets yields a decidable class with a nonconstant answer; a restriction of the plant does. If the automaton has no cycle labeled exclusively by events outside the shared alphabet, the two projections synchronize after a bounded number of events, both induced relations become regular over a finite alphabet of blocks, and OC and MOC become PSPACE-complete. Whether the boundedness assumption can be relaxed to the synchronicity of $\mathcal{R}_L$ remains open.

References

[1] W. M. Wonham and K. Cai, *Supervisory control of discrete-event systems*. Springer, 2019.

[2] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer, 2021.

[3] O. Boutin, J. Komenda, T. Masopust, K. Schmidt, and J. H. van Schuppen, "Hierarchical control with partial observations: Sufficient conditions," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 1817–1822.

[4] J. Komenda and T. Masopust, "Hierarchical supervisory control under partial observation: Normality," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7286–7298, 2023.

[5] S. Miao, J. Komenda, and F. Lin, "Hierarchical supervisory control of networked and cyber-attacked discrete-event systems," *Automatica*, vol. 183, p. 112578, 2026.

[6] J. Komenda and T. Masopust, "Supervisory control of modular discrete-event systems under partial observation: Normality," *IEEE Transactions on Automatic Control*, vol. 69, no. 6, pp. 3796–3807, 2024.

[7] S. Miao, J. Komenda, Y. Ji, and X. Yin, "Modular control for critical observability and weak opacity in discrete event systems," in *Proceedings of the 2026 American Control Conference (ACC)*, 2026.

[8] S. Hu, S. Miao, J. Komenda, and Z. Li, "Active prognosis and diagnosis of modular discrete-event systems," 2025.

[9] P. C. Fischer and A. L. Rosenberg, "Multitape one-way nonwriting automata," *Journal of Computer and System Sciences*, vol. 2, no. 1, pp. 88–101, 1968.

[10] A. Muscholl and G. Puppis, "The many facets of string transducers," in *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, ser. Leibniz International Proceedings in Informatics, vol. 126. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 2:1–2:21.

[11] J. Berstel, *Transductions and Context-Free Languages*, ser. Leitfäden der Angewandten Mathematik und Mechanik. Stuttgart: B. G. Teubner, 1979, vol. 38.

[12] J. Sakarovitch, *Elements of Automata Theory*. Cambridge University Press, 2009.

[13] L. J. Stockmeyer and A. R. Meyer, "Word problems requiring exponential time: Preliminary report," in *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1973, pp. 1–9.

[14] W. J. Savitch, "Relationships between nondeterministic and deterministic tape complexities," *Journal of Computer and System Sciences*, vol. 4, no. 2, pp. 177–192, 1970.

[15] D. Kozen, "Lower bounds for natural proof systems," in *18th Annual Symposium on Foundations of Computer Science (FOCS)*, 1977, pp. 254–266.