2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Process-Theoretic Approach to Supervisory Control Theory

J.C.M. Baeten, D.A. van Beek, B. Luttik, J. Markovski, and J.E. Rooda

***Abstract*— We revisit the central notion of controllability in supervisory control theory from process-theoretic perspective. To this end, we investigate partial bisimulation preorder, a behavioral preorder that is coarser than bisimulation equivalence and finer than simulation preorder. It is parameterized by a subset of the set of actions that need to be bisimulated, whereas the actions outside this set need only to be simulated. This preorder proves a viable means to define controllability in a nondeterministic setting as a refinement relation on processes. The new approach provides for a generalized characterization of controllability of nondeterministic discrete-event systems. We characterize the existence of a deterministic supervisor and compare our approach to existing ones in the literature. It helped identify the coarsest minimization procedure for nondeterministic plants that respects controllability. At the end, we define the notion of a maximally permissive supervisor, nonblocking property, and partial observability in our setting.**

I. Introduction

To keep products competitive, development costs and time-to-market need to be optimized, while satisfying ever-increasing demands for better quality, performance, safety, and ease of use. This puts high demands on the development of control software. Traditionally, software engineers write control software based on specification documents that contain informal requirements. This is a time-consuming process as the requirements are often ambiguous and they constantly change during product development. This issue in control software design gave rise to supervisory control theory of discrete-event systems [1], [2], where high-level supervisory controllers are synthesized automatically based upon formal models of the hardware and control requirements.

The supervisory controller observes the discrete-event behavior of the machine by receiving sensor signals from ongoing activities. Based upon these signals it makes a decision which activities are allowed to be carried out and sends back control signals to the hardware actuators. Under the assumption that the supervisory controller can react sufficiently fast on machine input, one can model this feedback loop as a pair of synchronizing processes. The model of the machine, referred to as *plant*, is restricted by the model of the controller, referred to as *supervisor*.

Traditionally, the plant is modeled as a set of observable traces of events, given as a set of synchronizing automata, whose joint recognized language corresponds to the observed traces. The events are split into *controllable events*, which can be disabled by the supervisor in the synchronous composition, and *uncontrollable events*, which must always be allowed by the supervisor. The *control requirements* specify allowed behavior again as sequences of events, leading to event-based supervisory control theory [1], [2].

In this paper, we model the feedback loop in a process-theoretic setting. We revisit the central notion of *controllability*, as constituted in [1], [2]. Controllability identifies sufficient and necessary conditions for existence of a supervisor for a given plant that satisfies the control requirements. Afterwards, we discuss related work and elaborate on the motivations and contributions of a process-theoretic approach.

A. Controllability

We introduce some preliminary notions of language theory [2]. Let $\mathcal{A} = \mathcal{C} \cup \mathcal{U}$ be the set of events that can be observed in the plant, with $\mathcal{C}$ being the set of controllable events and $\mathcal{U}$ the set of uncontrollable events, such that $\mathcal{C} \cap \mathcal{U} = \emptyset$. We form traces and languages in a standard manner, i.e., $t \in \mathcal{A}^*$ is a trace and $L \subseteq \mathcal{A}^*$ is a language, where $\mathcal{A}^* \triangleq \{a_1 a_2 \dots a_n \mid a_i \in \mathcal{A} \text{ for } 0 \leq i \leq n, n \in \mathbb{N}\}$ and ε denotes the unique empty trace $a_1 \dots a_n$ for $n = 0$. By $t{\cdot}t'$ we denote the concatenation of the traces $t, t' \in \mathcal{A}^*$ and by $L{\cdot}L' \triangleq \{t{\cdot}t' \mid t \in L, t' \in L'\}$ the concatenation of languages. We say that a language is prefix-closed if $L = \overline{L}$, where $\overline{L} \triangleq \{t \mid \text{there exists } t' \text{ such that } t{\cdot}t' \in L\}$. Suppose that $P = (\mathcal{S}, \mathcal{A}, \longmapsto, s_0)$ is a discrete-event automaton, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ a set of events, $\longmapsto \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ the transition relation, and s_0 the initial state. We define $\stackrel{*}{\longmapsto} \in \mathcal{S} \times \mathcal{A}^* \times \mathcal{S}$ as $s \stackrel{\varepsilon}{\longmapsto}{}^* s$ for all $s \in \mathcal{S}$, and $s \stackrel{at}{\longmapsto}{}^* s'$ for $a \in \mathcal{A}$ and $t \in \mathcal{A}^*$, if there exists $s'' \in \mathcal{S}$ such that $s \stackrel{a}{\longrightarrow} s'' \stackrel{t}{\longmapsto}{}^* s'$. By $s \stackrel{t}{\longmapsto}{}^*$ we denote that there exists $s' \in \mathcal{S}$ such that $s \stackrel{t}{\longmapsto}{}^* s'$. Now, the recognized (prefix-closed) language of automaton $P = (\mathcal{S}, \mathcal{A}, \longmapsto, s_0)$ is given by $L(P) \triangleq \{t \mid s_0 \stackrel{t}{\longmapsto}{}^* \}$. By $P_1 \mid P_2 \triangleq (\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}, \longmapsto, (s_1, s_2))$ we denote the synchronous parallel composition of $P_1 = (\mathcal{S}_1, \mathcal{A}, \longmapsto_1, s_1)$ and $P_2 = (\mathcal{S}_2, \mathcal{A}, \longmapsto_2, s_2)$, where $(s', s'') \stackrel{a}{\longmapsto} (\overline{s}', \overline{s}'')$ if $s' \stackrel{a}{\longmapsto}_1 \overline{s}'$ and $s'' \stackrel{a}{\longmapsto}_2 \overline{s}''$ for $s', \overline{s}' \in \mathcal{S}_1$, $s'', \overline{s}'' \in \mathcal{S}_2$, and $a \in \mathcal{A}$. We have that $L(P_1 \mid P_2) = L(P_1) \cap L(P_2)$.

Now, we define the property of controllability for prefix-closed languages. Suppose that the plant is given by automaton $P = (\mathcal{S}_P, \mathcal{A}, \longmapsto_P, s_P)$ and the control requirements by $R = (\mathcal{S}_R, \mathcal{A}, \longmapsto_R, s_R)$. An automaton $S = (\mathcal{S}_S, \mathcal{A}, \longmapsto_S, s_S)$ is a supervisor for P that achieves R if $L(P \mid S) = L(R)$, where we refer to $P \mid S$ as the *supervised plant*. We ensure that S does not disable uncontrollable events by requesting that R is *controllable* with respect to P, expressed as $L(R){\cdot}\mathcal{U} \cap L(P) \subseteq L(R)$ [1], [2]. Controllability is interpreted as follows. If we observe a desired trace in the plant followed by an uncontrollable event, then the control requirements cannot request that this uncontrollable

Eindhoven University of Technology P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands. Supported by C4C EU project (FP7-ICT-223844).

event should be disabled after allowing that trace. If R is controllable with respect to P, then one can guarantee the existence of a supervisor S, achieving the desired controlled behavior R by restricting the plant P, i.e., $L(P \mid S) = L(R)$.

In supervisory synthesis additional properties of $P \mid S$ are considered as well, e.g., notions of controllability that prevent deadlock and livelock. To this end, marked states are added to the automata to specify non-blocking behavior [1]. In our setting, we employ so-called *successful termination option* predicates [3]. Partial observability is another important property, where the assumption is that some events are hidden from the supervisor, e.g., due to lack of sensors [2]. Nonetheless, the supervisory controller must synchronize with the plant on unobservable events as well to achieve the desired behavior. We emphasize that in this paper we do not discuss supervisor synthesis algorithms and we investigate process-theoretic aspects of controllability.

B. Related Work

In a way, partial observability introduced nondeterminism in supervisory control theory. Nondeterministic automata are not disallowed in [1], but the semantics is still in terms of accepted languages. Nondeterminism naturally occurs in systems with multiple parallel components and it enables abstract (under)specifications and greater modeling convenience [3]. However, it introduces complications as controllability is originally a language-based property. This issue spawned investigations into the supervisory control of nondeterministic discrete-event systems.

In general, the supervisor is desired to be deterministic, as it should send unambiguous control signals and dutifully follow the state of the plant. An exception is [4], and references therein, where nondeterministic supervisors are considered under strong structural restrictions. State controllability is a notion tailored for such a setting [4], [5] and it requires all states of the control requirements reachable by a given trace to enable all outgoing uncontrollable events of states in the plant reachable by the same trace. Denote by $\mathrm{E}(s) \triangleq \{a \in \mathcal{A} \mid s \stackrel{a}{\longmapsto}\}$ the enabled events of s and by $\mathrm{E}^*(s,t) \triangleq \bigcup\{\mathrm{E}(s') \mid s \stackrel{t}{\longmapsto}^* s'\}$ the enabled events at all states reachable from s by the trace t. Then, control requirements R are *state controllable* with respect to a plant P, if for all $t \in L(R)$ and $r' \in \mathcal{S}_R$ such that $s_R \stackrel{t}{\longmapsto}^* r'$ it holds that $\mathrm{E}^*(s_P, t) \cap \mathcal{U} \subseteq \mathrm{E}(r')$.

State controllability induces language controllability in the deterministic case. Nonetheless, it is a restrictive notion since, e.g., a plant may not be state controllable with respect to itself, even though a supervisor enabling all events always exists [4]. We ponder on this issue in more depth later, and stipulate the need for state controllability. Other works tackle nondeterministic systems as a set of deterministic systems, by requiring controllability of all underlying deterministic systems to induce controllability of the nondeterministic system [6]. Nondeterminism is also modeled as a choice between unobservable events [7], hinting that the definition of state controllability might be inspired by partial observability.

An early approach that applies process theory to supervisory control synthesis is given in [8], where failure trajectories are employed and a CSP-like axiomatization of a specialized prioritized synchronization operator is given. Failure trajectories are extensions of failure semantics on whole traces, supporting compositionality of the prioritized synchronization that is employed to define controllability [8]. This operation is tailored to model the plant-supervisor communication and ensures that the supervisor cannot disable uncontrollable events. Followup works [7], [9] focus on deepening the understanding of the failure trajectories model and the prioritized synchronization. An alternative path is taken in [10], where instead of a new operator, a refinement relation $\ll$ based on failure semantics characterizes nondeterministic supervised behavior. For the automata P_1 and P_2 from above, $P_1 \ll P_2$ holds, if $L(P_1) \subseteq L(P_2)$, and for all $t \in L(P_1)$ it holds that $\mathcal{A} \setminus \mathrm{E}^*(s_1, t) \subseteq \mathcal{A} \setminus \mathrm{E}^*(s_2, t)$, where $\mathcal{A} \setminus \mathrm{E}^*(s_1, t)$ and $\mathcal{A} \setminus \mathrm{E}^*(s_2, t)$ are the unions of refusal sets of all states reachable in P_1 and P_2, respectively, following a trace t. Now, in addition to imposing language controllability, in [10] it is required that $P \mid S \ll R$ as well.

In [5] the refinement $\ll$ is given in terms of bisimulation (simulation in [11]), relying on state controllability. The use of (bi)simulation is also advocated in [12], [13], where nondeterminism arises due to inability of the controller to observe internal choices of the plant. Similarly to the approach of partial observability, all indistinguishable events are either always enabled or always disabled. There is no differentiation on controllable and uncontrollable events, and it is conjectured in [13] that some type of alternating (bi)simulation relation might be useful in such a setting.

C. Motivation and Contributions

A coalgebraic approach to supervisory control theory introduced *partial bisimulation* as a behavioral relation suitable to define controllability [14]. In essence, it states that controllable events should be simulated, whereas uncontrollable events should be bisimulated. It serves as a refinement relation between the supervised and the original plant, similar to the approach of [10], but for bisimulation semantics. Even though it is argued that refinements for failure and bisimulation semantics have similar properties [15], we consider (bi)simulation as a more elegant notion to capture nondeterminism [3], [16]. Refinements in failure semantics deal with traces and inclusion of refusal sets [10], whereas our notion relates states locally based on their outgoing transitions. Moreover, there exist efficient partitioning algorithms for minimization by (bi)simulation [17], already employed in the deterministic setting to optimize supervisor synthesis by imposing bisimulation over uncontrollable events [18].

Partial bisimulation is closely related to the notion of strong refinement of modal transition systems [19], where from each state there are so-called may and must transitions, corresponding to controllable (simulated) and uncontrollable (bisimulated) transitions. Then supervisor synthesis can also be seen as solving a process algebraic equation in the modal transition systems realm [20]. Nonetheless, refinement by

partial bisimulation is a special type of modal refinement, where the labels of the may and must transitions are fixed, admitting elegant process-algebraic characterization.

The contributions of this paper are as follows. First, we propose a process theory based on the preorder induced by partial bisimulation and we show some interesting properties of it and its induced equivalence. Using the obtained results we cast the control problem in a process-theoretic setting and we define a notion of a controllability using the partial bisimilarity preorder as a refinement between the supervised plant and the control requirements. The induced equivalence is basis for a minimization for (nondeterministic) plants that respects controllability. Furthermore, we characterize the existence of a deterministic supervisor given a nondeterministic plant and control requirements and relate it to similar notions in the literature. Finally, we discuss the existence of a maximally permissive supervisor and we cast nonblocking properties and partial observability in our setting. For technical details we refer to the supporting technical report [21].

II. Process Theory $\mathrm{BSP}_{|}(\mathcal{A}, \mathrm{B})$

We define a basic sequential process theory $\mathrm{BSP}_{|}(\mathcal{A}, \mathrm{B})$ with full synchronization and a partial bisimilarity preorder, following the nomenclature of [3]. The theory is parameterized with a finite set of actions $\mathcal{A}$ and *a bisimulation action set* $B \subseteq \mathcal{A}$, which plays a role in the behavioral relation.

The process terms $\mathcal{T}$ is induced by $P ::= 0 \mid 1 \mid a.P \mid P + P \mid P|P$ for $a \in \mathcal{A}$. The constant process 0 cannot execute any action and it can only deadlock, whereas 1 denotes the option to successfully terminate. The process corresponding to $a.p$ executes the action a and continues behaving as p. The alternative composition $p + q$ makes a nondeterministic choice by executing an action and continues to behave as the remainder of p or q. The synchronous parallel composition $p \mid q$ synchronizes all actions of p and q, and if no actions can be synchronized, it deadlocks.

We give semantics in terms of a successful termination option predicate $\downarrow \subseteq \mathcal{T}$ and a transition relation $\longrightarrow \subseteq \mathcal{T} \times \mathcal{L} \times \mathcal{T}$. We write $p\downarrow$ for $p \in \downarrow$ and $p \xrightarrow{a} p'$ for $(p, a, p') \in \longrightarrow$. We use the predicates $p \xrightarrow{a}$ and $p \not\xrightarrow{a}$ to denote that p has or does not have a transition labeled by a, respectively. We define $\downarrow$ and $\longrightarrow$ using structural operational semantics [3]:

$$\mathbf{1}\ \frac{}{1\downarrow} \quad \mathbf{2}\ \frac{p\downarrow}{p+q\downarrow} \quad \mathbf{3}\ \frac{q\downarrow}{p+q\downarrow} \quad \mathbf{4}\ \frac{p\downarrow,\ q\downarrow}{p \mid q\downarrow} \quad \mathbf{5}\ \frac{}{a.p \xrightarrow{a} p}$$

$$\mathbf{6}\ \frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \quad \mathbf{7}\ \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'} \quad \mathbf{8}\ \frac{p \xrightarrow{a} p',\ q \xrightarrow{a} q'}{p \mid q \xrightarrow{a} p' \mid q'}.$$

Next, we revisit the notion of the partial bisimulation [14].

Def. 1: A relation $R \subseteq \mathcal{T} \times \mathcal{T}$ is a partial bisimulation with respect to the bisimulation action set $B \subseteq \mathcal{A}$ if for all $p, q \in \mathcal{T}$ such that $(p, q) \in R$ it holds that:

1) if $p\downarrow$, then $q\downarrow$;
2) if $p \xrightarrow{a} p'$ for some $a \in \mathcal{A}$, then there exists $q' \in \mathcal{T}$ such that $q \xrightarrow{a} q'$ and $(p', q') \in R$;
3) if $q \xrightarrow{b} q'$ for some $b \in B$, then there exists $p' \in \mathcal{T}$ such that $p \xrightarrow{b} p'$ and $(p', q') \in R$.

We say that p is partially bisimilar to q with respect to the bisimulation action set B, notation $p \preceq_B q$, if there exists a partial bisimulation R with respect to B such that $(p, q) \in R$. If $q \preceq_B p$ holds as well, then p and q are mutually partially bisimilar with respect to B and we write $p \leftrightarrow_B q$. ■

Note that $\preceq_B$ is a preorder relation, making $\leftrightarrow_B$ an equivalence relation for all $B \subseteq \mathcal{A}$ [14]. If $B = \emptyset$, then $\preceq_\emptyset$ coincides with strong similarity preorder and $\leftrightarrow_\emptyset$ coincides with strong similarity equivalence [16], [3]. When $B = \mathcal{A}$, both $\preceq_\mathcal{A}$ and $\leftrightarrow_\mathcal{A}$ turn into strong bisimilarity [16], [3]. Moreover, if $p \preceq_B q$, then $p \preceq_C q$ for every $C \subseteq B$.

Thm. 1: Suppose $p \preceq_B q$ with $B \subseteq \mathcal{A}$ and $p, q \in \mathcal{T}$. Then: (1) $a.p \preceq_B a.q$; (2) $p + r \preceq_B q + r$ and $r + p \preceq_B r + q$; and (3) $p \mid r \preceq_B q \mid r$ and $r \mid p \preceq_B r \mid q$, for all $a \in \mathcal{A}$ and $r \in \mathcal{T}$. ■

Thm. 1 states that $\preceq_B$ is a precongruence, making $\leftrightarrow_B$ a congruence for $\mathcal{T}$ and providing for substitution rules. We build the term model $\mathbb{P}(\mathrm{BSP}_{|}(\mathcal{A}, \mathrm{B}))_{/\leftrightarrow_B}$ [3], where $\mathbb{P}(\mathrm{BSP}_{|}(\mathcal{A}, \mathrm{B})) = (\mathcal{T}, 0, 1, a.\_ \text{ for } a \in \mathcal{A}, \_ + \_, \_ \mid \_)$. The theory admits sound and ground-complete axiomatization for $\preceq_B$, whereas $\leftrightarrow_B$ is not finitely axiomatizable. Due to lack of space, we refer to [21] for technical details and extensions with recursion and modal characterization.

An important aspect of similarity-like equivalences, which plays an important role in their characterization are the so-called *little brother* terms [17], [22]. Their characterization makes possible a minimization procedure for mutual partial bisimilarity, which is the basis for plant aggregation that respects controllability. Two similar terms that do not contain little brothers are actually strongly bisimilar [22], implying the same property for partially bisimilar terms.

Def. 2: Let $p \xrightarrow{a} p'$ and $p \xrightarrow{a} p''$ for some $a \in \mathcal{A}$ and $p, p', p'' \in \mathcal{T}$. If $p' \preceq_B p''$ holds, but $p'' \preceq_B p'$ does not hold, then we say that p' is the little brother of p''. ■

The following equivalences shows how to eliminate little brothers provided that $p \preceq_B q \preceq_B r$ for $p, q, r \in \mathcal{T}$:

$$a.p + a.q \leftrightarrow_B a.q \qquad \text{if } a \notin B \qquad \text{LB1},$$
$$b.p + b.q + b.r \leftrightarrow_B b.p + b.r \qquad \text{if } b \in B \qquad \text{LB2}.$$

We note that LB1 is equivalent to the characteristic similarity relation $a.(p+q) + a.q \leftrightarrow_\emptyset a.(p+q)$ when $B = \emptyset$ [16]. Since the prefix action does not play a role in strong similarity, the relation there always holds. However, when the little brothers are prefixed by a bisimulation action $b \in B$, the 'littlest' and 'biggest' brother must be preserved, as given by LB2.

III. A Process-Theoretic Approach

We define controllability from a process-theoretic perspective in terms of partial bisimilarity preorder. We split $\mathcal{A}$ into a set of uncontrollable actions $\mathcal{U} \subseteq \mathcal{A}$, and a set of controllable actions $\mathcal{C} = \mathcal{A} \backslash \mathcal{U}$. We use $p \in \mathcal{T}$ to denote the plant and $r \in \mathcal{T}$ for the control requirements. The supervised plant is given by $p \mid s$ for a supervisor $s \in \mathcal{T}$. Intuitively, the uncontrollable transitions of the plant should be bisimilar to those of the supervised plant, so that the reachable uncontrollable part of the former is indistinguishable from that of the latter. The controllable transitions of the supervised plant may only be simulated by the ones of the original plant, since some controllable transitions are suppressed by the supervisor.

Def. 3: Let $p \in \mathcal{T}$ be a plant and $r \in \mathcal{T}$ control requirements. We say that $s \in \mathcal{T}$ is a supervisor for p that satisfies r if $p \mid s \preceq_{\mathcal{U}} p$ and $p \mid s \preceq_{\emptyset} r$. ■

As expected, Def. 3 ensures that no uncontrollable actions have been disabled in the supervised plant, by including them in the bisimulation action set. Moreover, it takes into account the nondeterministic behavior of the system. It suggests that the control requirements model the allowed behavior, independently of the plant. We opt for an 'external' specification in process-theoretic spirit and we require that the supervised plant has a behavior that is allowed, i.e., that can be simulated, by the control requirements.

This setting is also a preparation for future work, where we intend to relax the condition in the vein of [5], [11], abstracting in the control requirements from irrelevant internal actions, as advocated from process-theoretic perspective as well. Moreover, such an abstraction should preserve branching behavior, unlike the approach of [5], [11]. The goal in [5] is to achieve bisimilarity with the control requirements (similarity in [11]), again insinuating that the control requirements are seen as the (abstracted) desired behavior of the supervised plant to be achieved. The approach of [4] proposes a more closer coupling, requiring that the control requirements play the role of the supervisor as well.

If we assume that the control requirements coincide with the desired supervised behavior, i.e., $r \leftrightarrow_{\mathcal{U}} p \mid s$, then we only require that $r \preceq_{\mathcal{U}} p$, as $r \preceq_{\emptyset} r$ always holds, conforming to the original setting of [1]. Moreover, when p and r are deterministic, this coincides with language controllability, which was the original purpose of partial bisimilarity in [14].

Since we chose bisimilarity as an underlying notion that captures nondeterminism, one would expect that when we take the plant as the control requirements, the corresponding conditions $p \mid s \preceq_{\mathcal{U}} p$ and $p \mid s \preceq_{\emptyset} p$ would amount to bisimilarity. The conditions collapse to $p \mid s \preceq_{\mathcal{U}} p$, since $p \mid s \preceq_{\mathcal{U}} p$ implies $p \mid s \preceq_{\emptyset} p$. Now, we seek the largest possible supervised plant, i.e., $p \preceq_{\mathcal{U}} p \mid s$, leading to $p \mid s \leftrightarrow_{\mathcal{U}} p$. Note, however, that the plant may have redundant behavior in the form of little brothers, which prevents bisimilarity between p and $p \mid s$. By eliminating the little brothers using LB1 and LB2, we have that $p \mid s \leftrightarrow_{\mathcal{U}} p$ implies $p \mid s \leftrightarrow_{\mathcal{A}} p$ [22].

A. State Controllability and Nondeterministic Supervisors

Relating our notion to state controllability [4], [5], it is known that some plants are not state controllable when the control requirements coincide with the plant, even though a trivial supervisor that enables all events always exists. For instance, let p and r coincide with $p \triangleq u.v.0 + u.w.0$, where $\mathcal{U} = \{u, v, w\}$. Then the enabled uncontrollable events following the trace u are given by $\mathrm{E}^*(p, u) = \{v, w\}$ (here we overload the definitions of E^* and E from the introduction). Following the same trace in the control requirements, we reach $r \xrightarrow{u} v.0$ or $r \xrightarrow{u} w.0$ with $\mathrm{E}(v.0) = \{v\}$ and $\mathrm{E}(w.0) = \{w\}$. Since, $\{v, w\} \cap \mathcal{U} \not\subseteq \{v\}$, we conclude that the plant is not state controllable with respect to itself. However, a non-restrictive supervisor $s \triangleq u.(v.0 + w.0)$, induced by the determinized version of the plant, always exists. This is supported by Def. 3, since when $p \mid s$ coincides with p, we trivially have that $p \preceq_{\mathcal{U}} p$ and $p \preceq_{\emptyset} p$, implying that s is a supervisor for p that satisfies p.

However, a truly nondeterministic supervisor, i.e., one having a choice between two transitions labeled by u that do not lead to partially bisimilar states, does not exist. To illustrate, the minimal nondeterministic supervisor s' is given by the plant itself, i.e., $s' \triangleq p$. We have $p \mid s' \leftrightarrow_{\mathcal{U}} u.0 + u.v.0 + u.w.0$ implying that $p \mid s' \preceq_{\mathcal{U}} p$ does not hold. We conclude that state controllability is not a suitable characterization of an existence of a deterministic supervisor for a nondeterministic plant and control requirements.

Def. 3 also admits nondeterministic supervisors in the vein of [4], [5]. As an illustration, suppose that $p \triangleq a.(b.0 + c.0)$ and $r \triangleq a.b.0 + a.c.0$ with $\mathcal{C} = \{a, b, c\}$. Obviously, a deterministic supervisor that achieves r does not exist, whereas a nondeterministic supervisor s that coincides with r, i.e., $s \triangleq a.b.0 + a.c.0$, trivially satisfies both state controllability, as there are no uncontrollable events, and Def. 3, as $p \mid s \leftrightarrow_{\mathcal{A}} r$ and $r \preceq_{\emptyset} p$. Intuitively, nondeterministic supervisors increase plant nondeterminism in the sense that they increase the number of states with nondeterministic choices that are reachable by the same trace. In the literature [4], [7], [5], [11], this is needed in order to satisfy some nondeterministically weaker control requirements as in the example above.

B. Process-Theoretic Definition of Controllability

As illustrated above, a usual suspect for a deterministic supervisor is the determinized version of the desired supervised behavior. We define a determinized process $\det(p) \in \mathcal{T}$ as the minimal process that enables all possible traces of $p \in \mathcal{T}$:

$$\mathbf{9}\ \frac{p\!\downarrow}{\det(p)\!\downarrow} \qquad \mathbf{10}\ \frac{p \xrightarrow{a}}{\det(p) \xrightarrow{a} \det(\sum\{p' \in \mathcal{T} \mid p \xrightarrow{a} p'\})}$$

Rule 9 states that the original and determinized version of a process have the same termination options. Rule 10 merges a nondeterministic choice over equally labeled transitions to a single transition modulo bisimilarity, of which the target is the alternative composition of all original target processes modulo commutativity and associativity. For example, suppose that the only outgoing transitions of p that are labeled by a are $p \xrightarrow{a} p'$ and $p \xrightarrow{a} p''$. Then, $\det(p) \xrightarrow{a} p' + p''$ and $\det(p) \xrightarrow{a} p'' + p'$, and clearly $p' + p'' \leftrightarrow_{\mathcal{A}} p'' + p'$. Now, we can define a *deterministic* process to be one that is bisimilar to its determinized version, i.e., p is deterministic if $p \leftrightarrow_{\mathcal{A}} \det(p)$. Clearly, all determinized processes are deterministic.

Thm. 2: For all $p, q \in \mathcal{T}$ it holds that (1) $p \mid \det(p) \leftrightarrow_{\mathcal{A}} p$ and (2) if $p \preceq_B q$ then $\det(p) \mid q \preceq_B q$ for $B \subseteq \mathcal{A}$. ■

Property (1) states that the synchronization of a process with its determinized version does not restrict its behavior. If two processes are partially bisimilar, then their determinized versions are partially bisimilar as well, as stated by property (2). Note that the other direction does not hold in general.

Now, suppose that the desired supervised behavior is given by $q \in \mathcal{T}$. It can be achieved if there exists a supervisor $s \in \mathcal{T}$, such that $p \mid s \leftrightarrow_{\mathcal{U}} q$. Since Def. 3 requires that

$p \mid s \preceq_{\mathcal{U}} p$ and $p \mid s \preceq_{\emptyset} r$, we have that $q \preceq_{\mathcal{U}} p$ and $q \preceq_{\emptyset} r$ are necessary conditions. As discussed above, a good supervisor candidate is $s \triangleq \det(q)$, since from $q \preceq_{\mathcal{U}} p$ we have that $q \mid \det(q) \preceq_{\mathcal{U}} p \mid \det(q)$, implying $q \preceq_{\mathcal{U}} p \mid \det(q)$ using property (1) of Thm. 2. Furthermore, according to property (2) of Thm. 2 we have that $p \mid \det(q) \preceq_{\mathcal{U}} p$. Next, we characterize when a desired behavior is controllable.

Def. 4: Process $q \in \mathcal{T}$ is controllable with respect to plant $p \in \mathcal{T}$ and control requirements $r \in \mathcal{T}$, if $q \preceq_{\mathcal{U}} p$, $q \preceq_{\emptyset} r$, and $p \mid \det(q) \preceq_{\mathcal{U}} q$. ■

Def. 4 requires that the plant partially bisimulates and the control requirements simulate the supervised behavior. This ensures that Def. 3 is satisfied. By property (2) of Thm. 2, this implies that the deterministic behavior of the supervised plant, i.e., its language, is partially bisimilar to the plant. Thus, the supervised behavior is language-controllable with respect to plant, fortifying it as a choice for a deterministic supervisor. In return, it partially bisimulates the supervised plant, lifting the notion of language closure [1] and implying that they are mutually partially bisimilar.

Thm. 3: If $q \in \mathcal{T}$ is controllable with respect to a plant $p \in \mathcal{T}$ and control requirements $r \in \mathcal{T}$, then $\det(q)$ is a supervisor for p with respect to r such that $p \mid \det(q) \leftrightarrow_{\mathcal{U}} q$. ■

The minimal deterministic supervisor s for p such that $p \mid s$ contains the behavior of q, i.e., $q \preceq_{\mathcal{U}} p \mid s$, is $s = \det(q)$. So, for any other supervisor $\det(s') \in \mathcal{T}$ we must have that $\det(q) \preceq_{\emptyset} \det(s')$ and $p \mid \det(s') \preceq_{\mathcal{U}} p$. Furthermore, we can also demand that the control requirements r are controllable. In this case, the conditions of Def 4 amount to $r \preceq_{\mathcal{U}} p$ and $p \mid \det(r) \preceq_{\mathcal{U}} r$, comparable to the approaches of [1], [10], [5], [11], [12], [13]. For deterministic systems, the first condition of Def. 4 coincides with language controllability of [1], as shown in [14].

Finally, satisfiability of the requirements can be efficiently checked using an algorithm that computes the mutual partial bisimilarity quotient, see [21]. Moreover, we can replace p by every $p' \in \mathcal{T}$ such that $p' \leftrightarrow_{\mathcal{U}} p$. Thus, minimization by mutual partial bisimilarity provides for the coarsest plant that preserves controllability, a notion lacking in previous work.

To relate more closely our notion to state controllability, we reformulate Def. 4 in terms of traces. Assuming that $q \preceq_{\mathcal{U}} p$, the existence of a supervisor depends on whether $p \mid \det(q) \preceq_{\mathcal{U}} q$. In terms of traces, we require that for every trace $t = a_1 a_2 \dots a_n \in \mathcal{A}^*$ and every $p_n \in \mathcal{T}$ such that $p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n$, there exist $q_1, \dots, q_n \in \mathcal{T}$ such that $q \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n$ and $\mathrm{E}(p_i) \cap \mathrm{E}^*(\det(q), t_i) \subseteq \mathrm{E}(q_i)$ and $\mathrm{E}(p_i) \cap \mathcal{U} = \mathrm{E}(q_i) \cap \mathcal{U}$ with $t_i = a_1 \dots a_i$ for $i \in \{1, \dots, n\}$. Recall that state controllability requires that every state of q has to be able to 'simulate' the uncontrollable behavior of all states of p reachable by the same trace. In contrast, our notion requires the same uncontrolled behavior only for related states of p and q that are reachable by the same trace. We observe, however, from the reformulation that when resorting to truly nondeterministic supervisors, the above must hold for every trace of the supervisor, ultimately amounting to state controllability.

C. Maximal Permissiveness, Nonblocking Property, and Partial Observability

When the desired supervised behavior is not achievable, in the sense that every other achievable supervised behavior is partially bisimilar to the maximal permissive one, we have to resort to the notion of maximally permissive supervisors [1], [2]. In the language setting, the maximal permissive behavior is achieved as a union of the languages of all possible controllable behaviors. Here, the role of the union is taken by the alternative composition that introduces additional traces. Suppose that $q = q_1 + q_2$, where both q_1 and q_2 are controllable. Then, according to Def. 4, we have that $p \mid \det(q_1) \leftrightarrow_{\mathcal{U}} q_1$ and $p \mid \det(q_2) \leftrightarrow_{\mathcal{U}} q_2$, i.e., deterministic supervisors $\det(q_1)$ and $\det(q_2)$ exist. It follows that $p \mid (\det(q_1) + \det(q_2)) \leftrightarrow_{\mathcal{U}} q_1 + q_2$. However, for $q_1 + q_2$ to be controllable, it must be that $p \mid \det(q_1 + q_2) \leftrightarrow_{\mathcal{U}} q_1 + q_2$. Thus, we need $p \mid \det(q_1 + q_2) \preceq_{\mathcal{U}} p \mid (\det(q_1) + \det(q_2))$, since $p \mid (\det(q_1) + \det(q_2)) \preceq_{\mathcal{U}} p \mid \det(q_1 + q_2)$ always holds. The former relation characterizes when maximal permissiveness of two controllable processes is achievable. Accordingly, we can define a *maximally permissive supervised plant* given a plant p and control requirements r as $q^{\uparrow C} \triangleq \sum\{q \in \mathcal{T} \mid q \text{ is controllable with respect to } p \text{ and } r\}$, provided that $p \mid \det(q_1 + q_2) \preceq_{\mathcal{U}} p \mid (\det(q_1) + \det(q_2))$ for all $q_1, q_2 \in \mathcal{T}$ that are controllable with respect to p and r.

It is not difficult to show that when the plant and the control requirements are deterministic, every controllable behavior is deterministic as well, and the above requirements is satisfied. Thus, in the deterministic case, there always exists a maximally permissive supervised behavior, conforming to [1], provided that the minimal supervised plant behavior with respect to the partial bisimilarity preorder $\preceq_{\mathcal{U}}$ is controllable. According to Def. 3, the minimal supervised plant is the initial uncontrollable reach of the plant, i.e., the reachable part of the plant by taking only uncontrollable prefixes. For example, the minimal supervised behavior of $p \triangleq u.v.0 + c.u.0 + v.c.0$, with $\mathcal{U} = \{u, v\}$ and $\mathcal{C} = \{c\}$, is $u.v.0 + v.0$. The deadlock process can be taken as the minimal supervised behavior only if the initial state of the plant does not have outgoing uncontrollable transitions.

Next, we remark that the non-blocking property of [1], [2] can be specified in our setting as a reachability property. If we suppose that some states in the plant automaton P are defined as marked, given by the set M, we can define the marked language of P as $L_M = \{t \in \mathcal{A}^* \mid \text{ there exists } s' \in M \text{ such that } s \overset{t}{\longmapsto}{}^*_P s'\}$. Then, the supervised plant is nonblocking if $\overline{L_m(P \mid S)} = L(P \mid S)$, i.e., we can extend every trace with a trace that ends in a marked state [1], [2]. To this end, we can employ the successful termination predicate and denote a 'state' $p \in \mathcal{T}$ to be marked if $p{\downarrow}$. Then a given controllable supervised behavior q is *nonblocking* if for every $q' \in \mathcal{T}$ such that $q \xrightarrow{t}{}^* q'$ for some $t \in \mathcal{A}^*$, there exists $q'' \in \mathcal{T}$ and $t' \in \mathcal{A}^*$ such that $q' \xrightarrow{t'}{}^* q''$ and $q''{\downarrow}$.

Finally, we cast the notion of partial observability in our setting [2]. In supervision under partial observability it is assumed that not all events are observable by the supervisor.

They are split to observable events $\mathcal{O} \subseteq \mathcal{A}$ and unobservable events $\mathcal{A} \setminus \mathcal{O}$. Partial observability is a global property that states that in all states of the control requirements that reachable by the same *observable* trace, an observable event that is also allowed in the plant following that trace, must be either always enabled or disabled. The difficulty in capturing this property in a process-theoretic setting lies in the fact that the states that are reachable by the same trace do not have to be otherwise related. An attempt was made in [23] to capture this notion as a separate state-partitioning relation that was later coupled to controllability. Here, we will rely on a set of relevant states of the control requirements to keep track that events in all states are enabled or disabled.

Def. 5: A relation $R \subseteq \mathcal{T} \times \mathcal{T} \times 2^{\mathcal{T}}$ is a partial bisimulation with partial observability with respect to the bisimulation action set $B \subseteq \mathcal{A}$ and observable action set $\mathcal{O} \subseteq \mathcal{A}$ if for all $p, q \in \mathcal{T}$ and $\Omega \subset \mathcal{T}$ such that $(p, q, \Omega) \in R$ it holds that:

1) if $p\!\downarrow$, then $q\!\downarrow$;
2) if $p \xrightarrow{a} p'$ for some $a \in \mathcal{O}$, then there exist $q' \in \mathcal{T}$ and $\Omega' \subset \mathcal{T}$ such that $q \xrightarrow{a} q'$ and $\bar{p} \xrightarrow{a}$ for all $\bar{p} \in \Omega$, and $\Omega' = \{\bar{p}' \mid \bar{p} \xrightarrow{a} \bar{p}', \bar{p} \in \Omega\}$ with $(p', q', \Omega') \in R$;
3) if $p \xrightarrow{a} p'$ for some $a \notin \mathcal{O}$, then there exist $q' \in \mathcal{T}$ and $\Omega' \subset \mathcal{T}$ such that $q \xrightarrow{a} q'$ and $\Omega' = \Omega \cup \{\bar{p}' \mid \bar{p} \xrightarrow{a} \bar{p}', \bar{p} \in \Omega\}$ with $(p', q', \Omega') \in R$;
4) if $q \xrightarrow{b} q'$ for some $b \in B \cap \mathcal{O}$, then there exist $p' \in \mathcal{T}$ and $\Omega' \subset \mathcal{T}$ such that $p \xrightarrow{b} p'$ and $\bar{p} \xrightarrow{b}$ for all $\bar{p} \in \Omega$, and $\Omega' = \{\bar{p}' \mid \bar{p} \xrightarrow{b} \bar{p}', \bar{p} \in \Omega\}$ with $(p', q', \Omega') \in R$;
5) if $q \xrightarrow{b} q'$ for some $b \in B \setminus \mathcal{O}$, then there exist $p' \in \mathcal{T}$ and $\Omega' \subset \mathcal{T}$ such that $p \xrightarrow{b} p'$ and $\Omega' = \Omega \cup \{\bar{p}' \mid \bar{p} \xrightarrow{b} \bar{p}', \bar{p} \in \Omega\}$ with $(p', q', \Omega') \in R$. ■

The set Ω in Def. 5 keeps track of all states of the control requirements that can be reached by the same observable trace as the current state and it ensures that all observable actions are also available for all reachable states as they are simulated by the plant. Given a plant p and a desired supervised behavior q, we require that there exists a partial bisimulation with partial observability relation R such that $(q, p, \{q\}) \in R$ to ensure that no uncontrollable events are disabled and partial observability is retained.

IV. Concluding Remarks

We successfully employed partial bisimilarity preorder to define controllability of nondeterministic processes. Our definition is finer than existing notions in the literature and it reduces to language controllability for deterministic systems. To support this investigation we developed a process theory in which we casted standard notion from supervisory control theory. Furthermore, we characterized the existence of a deterministic supervisor and a maximally permissive supervised behavior, and we discussed the relation with other notions in the literature. Our investigation identified minimization by mutual partial bisimilarity as the coarsest controllability-preserving minimization.

As future work, we aim to improve existing algorithms for supervisor synthesis based on the obtained insights and apply them to existing case studies. Further on, we plan to apply the prominent process-theoretic techniques of abstraction and hiding to supervisory control. Other interesting topics are modular control, as concurrency is dealt with elegantly in process algebra, as well as extensions with quantitative aspects like time or probabilities.

References

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

[2] C. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Kluwer Academic Publishers, 2004.

[3] J. C. M. Baeten, T. Basten, and M. A. Reniers, *Process Algebra: Equational Theories of Communicating Processes*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2010, vol. 50.

[4] M. Fabian and B. Lennartson, "On non-deterministic supervisory control," *Proceedings of the 35th IEEE Decision and Control*, vol. 2, pp. 2213–2218, 1996.

[5] C. Zhou, R. Kumar, and S. Jiang, "Control of nondeterministic discrete-event systems for bisimulation equivalence," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 754–765, 2006.

[6] S.-J. Park and J.-T. Lim, "Nonblocking supervisory control of nondeterministic systems based on multiple deterministic model approach," *IEICE Transactions on Information and Systems*, vol. E83-D, no. 5, pp. 1177–1180, 2000.

[7] M. Heymann and F. Lin, "Discrete-event control of nondeterministic systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 3–17, 1998.

[8] M. Heymann and G. Meyer, "Algebra of discrete event processes," NASA Ames Research Center, Technical Report NASA 102848, 1991.

[9] R. Kumar and M. A. Shayman, "Nonblocking supervisory control of nondeterministic systems via prioritized synchronization," *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1160–1175, 1996.

[10] A. Overkamp, "Supervisory control using failure semantics and partial specifications," *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 498–510, 1997.

[11] R. Kumar and C. Zhou, "Control of nondeterministic discrete event systems for simulation equivalence," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pp. 340–349, 2007.

[12] P. Madhusudan and P. S. Thiagarajan, "Branching time controllers for discrete event systems," *Theoretical Computer Science*, vol. 274, no. 1-2, pp. 117–149, 2002.

[13] P. Tabuada, "Controller synthesis for bisimulation equivalence," *Systems and Control Letters*, vol. 57, no. 6, pp. 443–452, 2008.

[14] J. J. M. M. Rutten, "Coalgebra, concurrency, and control," Center for Mathematics and Computer Science, Amsterdam, The Netherlands, SEN Report R-9921, 1999.

[15] R. Eshuis and M. M. Fokkinga, "Comparing refinements for failure and bisimulation semantics," *Fundamenta Informaticae*, vol. 52, no. 4, pp. 297–321, 2002.

[16] R. J. v. Glabbeek, "The linear time–branching time spectrum I," *Handbook of Process Algebra*, pp. 3–99, 2001.

[17] R. Gentilini, C. Piazza, and A. Policriti, "From bisimulation to simulation: Coarsest partition problems," *Journal of Automated Reasoning*, vol. 31, no. 1, pp. 73–103, 2003.

[18] G. Barrett and S. Lafortune, "Bisimulation, the supervisory control problem and strong model matching for finite state machines," *Discrete Event Dynamic Systems*, vol. 8, no. 4, pp. 377–429, 1998.

[19] K. G. Larsen, "Modal specifications," in *Automatic Verification Methods for Finite State Systems*, ser. LNCS, vol. 407. Springer, 1990, pp. 232–246.

[20] K. G. Larsen and L. Xinxin, "Equation solving using modal transitions systems," in *Proceedings of LICS*. IEEE, 1990, pp. 108–117.

[21] J. C. M. Baeten, D. A. van Beek, B. Luttik, J. Markovski, and J. E. Rooda, "Partial bisimulation," Eindhoven University of Technology, SE Report 10-04, 2010, available from `http://se.wtb.tue.nl`.

[22] C. Baier and J.-P. Katoen, *Principles of Model Checking*. MIT Press, 2008.

[23] J. Komenda and J. H. van Schuppen, "Control of discrete-event systems with partial observations using coalgebra and coinduction," *Discrete Event Dynamic Systems*, vol. 15, pp. 257–315, 2005.