Optimal Supervisory Control of Probabilistic Discrete Event Systems

Vera Pantelic and Mark Lawford

Abstract

This paper considers optimal supervisory control of probabilistic discrete event systems (PDESs). PDESs are modeled as generators of probabilistic languages. The probabilistic supervisors employed enable/disable events with certain probabilities. We consider the case when there exists no probabilistic supervisor to match the behaviour of a plant to a probabilistic requirements specification. First, we define a notion of distance between two probabilistic generators. Then, given a plant and a desired probabilistic behaviour, we present an algorithm that minimizes the distance between the desired behaviour and the behaviour of the controlled plant achievable under probabilistic control.

Software Quality Research Laboratory

SQRL Report No. 55

DRAFT

Authors are with the SQRL, Department of Computing and Sofware, Faculty of Engineering, McMaster University, 1280 Main Street West, Hamilton ON, L8S 4K1, Canada `pantelv at mcmaster dot ca, lawford at mcmaster dot ca`

I. INTRODUCTION

In order to model stochastic behaviour of a plant, many models of stochastic behavior of discrete event systems have been proposed (e.g., Markov chains [1], Rabin's probabilistic automata [2], stochastic Petri nets [3]). We follow the theory of probabilistic DES that was developed in [4], [5] using an algebraic approach. A stochastic discrete event system is represented as an automaton with transitions labeled with probabilities. As opposed to the model of [4], [5], our probabilistic automaton is deterministic in the following sense: for each state of the automaton and any event, there is at most one next state to which the automaton can move. The probabilities of all the events in a certain state add up to at most one. With Rabin's probabilistic automata [2], on the other hand, the sum of the transition probabilities of an event at a state is one. Also, unlike the Markov chains [1], the emphasis of the approach of [4], [5] is on event traces rather than state traces.

The control of different models of stochastic discrete event systems has been investigated in [6], [7], etc. Rabin's probabilistic automata are used in [6] as the underlying model, while [7] investigates the optimal control theory of Markov chains. A deterministic supervisory control framework for stochastic discrete event systems was developed in [8] using the model of [4], [5]. Controllable events are disabled dynamically as first suggested in [9], so that the probabilities of their execution become zero, and the probabilities of the occurrence of other events proportionally increase. The control objective considered in [8] is to construct a supervisor such that the controlled plant does not execute specified illegal traces, and occurrences of the events in the system are greater or equal to specified values. The paper gives necessary and sufficient condition for the existence of a supervisor. Further, in [10], a technique to compute a maximally permissive supervisor on-line is given. In [11], [12], the probabilistic generators introduced in [4], [5] are used. The requirements specification is given by weights assigned to states of a plant and the control goal is, roughly speaking, to reach the states with more weight (more desired states) more often. A deterministic control is synthesized for a given requirements specification so that a measure based on the specification and probabilities of the plant is optimized.

In [9], PDESs are modeled as probabilistic generators adopted from [4], [5], and deterministic supervisors for DES are generalized to *probabilistic supervisors*. The probabilistic supervisors are so named because they employ the control method of *random disablement*: after observing a string s, the probabilistic supervisor enables an event σ with a certain probability. Standard (deterministic) control can deterministically enable/disable controllable events. As shown in [9], a plant under probabilistic control can generate a larger class of probabilistic languages than deterministic control. The supervisory control problem considered is to find, if possible, a supervisor under whose control the behaviour of a plant is identical to a given probabilistic specification. The necessary and sufficient conditions for the existence of a supervisor for a class of PDESs are given in [9]. A formal proof of the necessity and sufficiency of the conditions and an algorithm for the calculation of the supervisor are presented in [13], [14].

Controller synthesis for probabilistic systems has also attracted attention in the formal methods community. E.g., [15], [16] consider different control policies: deterministic or randomized (probabilistic) on one hand; memoryless (Markovian) or history-independent on the other. The systems considered are finite Markov decision processes, where the state space is divided into two disjoint sets: controllable states and uncontrollable states. In [15], the controller synthesis problem for a requirements specification given as a probabilistic computation tree logic (PCTL) formula is shown to be NP-hard, and a synthesis algorithm for automata specification is presented. Controller synthesis was considered in [16] for requirements specification given as a formula of PCTL extended with long-run average propositions. It is shown that the existence of such a controller is decidable, and an algorithm for the synthesis of a controller, when it exists, is presented. Further, controller robustness with respect to slight changes in the probabilities of the plant is discussed. The paper shows that the existence of robust controllers is decidable and the controller, if it exists, is effectively computable.

Deterministic control is easier to deal with than probabilistic control, both from the viewpoint of analysis, and practice. However, probabilistic control is much more powerful. It has been shown in [9] that probabilistic supervisory control can generate a much larger class of probabilistic languages than deterministic control. In the sense of the supervisory control problem discussed in this paper, the use of deterministic control might be too restrictive for a designer. Hence, [9] and [13] investigate probabilistic supervisory control: conditions under which a probabilistic control can generate a prespecified language, and, if the supervisor exists, the algorithm for its synthesis.

Analogous to a problem in classical supervisory control theory, it can happen that, given a plant to be controlled and a probabilistic specification language, no supervisor exists such that the plant under control generates the pre-specified language. In this case, when the exact solution is not achievable, a designer tries to find a supervisor such that the plant generates the behavior closest to the desired behaviour. The measure of proximity in our case will be a pseudometric on the states of probabilistic transition systems.

In Section II we present PDES as generators of probabilistic languages, and introduce the probabilistic control of PDES. The proposed pseudometric is presented in Section III. Its characterization as the greatest fixed point of a function (as presented in [17]) is used for the derivation and proof of the correctness of two algorithms for the calculation of the distances between the states of a PDES in this pseudometric. Section IV presents the algorithm for finding the closest approximation to within a prespecified accuracy. Section V concludes with avenues for future work.

Results of this research were first published in [18], lacking much of informal reasoning or formal proofs of algorithms presented. In this paper we give a more formal and detailed version of that work.

II. Preliminaries

In this section, we present PDES modeled as generators of probabilistic languages. Then, we introduce the probabilistic control of PDESs, the probabilistic supervisory problem, and the main results of [9], [13].

A. Modeling PDES

The probabilistic DES can be modeled as a probabilistic generator $G = (Q, \Sigma, \delta, q_0, Q_m, p)$ [9], where Q is the nonempty set of states (at most countable), Σ is a finite alphabet whose elements we will refer to as event labels, $\delta : Q \times \Sigma \to Q$ is the (partial) transition function, $q_0 \in Q$ is the initial state, $Q_m \subseteq Q$ is the set of marking states, which represent the completed tasks, and $p : Q \times \Sigma \to [0, 1]$ is the statewise event probability distribution. The state transition function is traditionally extended by induction on the length of strings to $\delta : Q \times \Sigma^* \to Q$:

$$\begin{aligned} \delta(q, \epsilon) &= q, \\ \delta(q, s\sigma) &= \delta(\delta(q, s), \sigma), \quad q \in Q, s \in \Sigma^*, \sigma \in \Sigma \end{aligned}$$

whenever $\delta(q, s)$ and $\delta(\delta(q, s), \sigma)$ are both defined. For a state q, and a string s, the expression $\delta(q, s)!$ will denote that δ is defined for the string s in the state q.

The probability that the event $\sigma \in \Sigma$ is going to occur at the state $q \in Q$ is $p(q, \sigma)$. For the generator G to be well-defined, (i) $p(q, \sigma) = 0$ should hold if and only if $\delta(q, \sigma)$ is undefined, and (ii) $\forall q \ \sum_{\sigma \in \Sigma} p(q, \sigma) \leq 1$. The probabilistic generator G is nonterminating if, for every reachable state $q \in Q$, $\sum_{\sigma \in \Sigma} p(q, \sigma) = 1$. Conversely, G is terminating if there is at least one reachable state $q \in Q$ such that $\sum_{\sigma \in \Sigma} p(q, \sigma) < 1$. The probability that the system terminates at state q is $1 - \sum_{\sigma \in \Sigma} p(q, \sigma)$. Throughout the sequel, we will mostly consider nonterminating generators.

The language $L(G)$ generated by a probabilistic DES automaton $G = (Q, \Sigma, \delta, q_0, Q_m, p)$ is $L(G) = \{s \in \Sigma^* \mid \delta(q_0, s)!\}$. The marked language of G is $L_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$, whereas the probabilistic language generated by G is defined as:

$$L_p(G)(\epsilon) = 1$$

$$L_p(G)(s\sigma) = \begin{cases} L_p(G)(s) \cdot p(\delta(q_0, s), \sigma) & \text{if } \delta(q_0, s)! \\ 0 & \text{otherwise} \end{cases}$$

Informally, $L_p(G)(s)$ is the probability that the string s is executed in G. Also, $L_p(G)(s) > 0$ iff $s \in L(G)$.

For each state $q \in Q$, we define the function $\rho_q : \Sigma \times Q \to [0, 1]$ such that for any $q' \in Q$, $\sigma \in \Sigma$, we have $\rho_q(\sigma, q') = p(q, \sigma)$ if $q' = \delta(q, \sigma)$, and 0 otherwise. The function ρ_q is a probability distribution on the set $\Sigma \times Q$. Also, for a state q, we define *the set of possible events* to be $Pos(q) := \{\sigma \in \Sigma | p(q, \sigma) > 0\}$.

B. Probabilistic Supervisors: Existence and Synthesis

As in classical supervisory theory, the set Σ is partitioned into Σ_c and Σ_u, the sets of controllable and uncontrollable events, respectively. Deterministic supervisors for DES are generalized to *probabilistic supervisors*. The control technique used is called *random disablement*. Instead of deterministically enabling or disabling controllable events, probabilistic supervisors enable them with certain probabilities. This means that, upon reaching a certain state q, the control pattern is chosen according to supervisor's probability distributions of controllable events. Consequently, the controller does not always enable the same events when in the state q.

For a PDES $G = (Q, \Sigma, \delta, q_0, Q_m, p)$, a *probabilistic supervisor* is a function $V_p : L(G) \times \Sigma \to [0, 1]$ such that

$$(\forall s \in L(G))(\forall \sigma \in \Sigma)V_p(s, \sigma) = \begin{cases} 1 & \text{if } \sigma \in \Sigma_u \\ x_{s,\sigma} \in [0, 1] & \text{otherwise.} \end{cases}$$

Therefore, after observing a string s, the supervisor enables the event σ with probability $V_p(s, \sigma)$. After a set of controllable events to be enabled has been decided upon (uncontrollable events are always enabled), the system acts as if supervised by a deterministic supervisor.

The goal is to match the behaviour of the controlled plant with a given probabilistic specification language. We call this problem the *Probabilistic Supervisory Control Problem (PSCP)*. More formally:

Given a plant PDES G_1 and a specification PDES G_2, find, if possible, a probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$.

We now present the conditions for the existence of a probabilistic supervisor for PSCP (that were first presented in [9]), and an algorithm for the computation of the supervisor which is the main result of [13].

For notational convenience, instead of $x(\sigma)$ ($x(\sigma) \in \mathbb{R}, \sigma \in \Sigma$), we will write x_σ.

Theorem 1: Let $G_1 = (Q, \Sigma, \delta_1, q_0, Q_m, p_1)$ and $G_2 = (R, \Sigma, \delta_2, r_0, R_m, p_2)$ be two nonterminating PDESs with disjoint state sets Q and R. There exists a probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$ iff for all $s \in L(G_2)$ there exists $q \in Q$ such that $\delta_1(q_0, s) = q$ and, letting $r = \delta_2(r_0, s)$, the following two conditions hold:

(i) $Pos(q) \cap \Sigma_u = Pos(r) \cap \Sigma_u$, and for all $\sigma \in Pos(q) \cap \Sigma_u$,

$$\frac{p_1(q, \sigma)}{\sum\limits_{\alpha \in \Sigma_u} p_1(q, \alpha)} = \frac{p_2(r, \sigma)}{\sum\limits_{\alpha \in \Sigma_u} p_2(r, \alpha)}$$

(ii) $Pos(r) \cap \Sigma_c \subseteq Pos(q) \cap \Sigma_c$, and, if $Pos(q) \cap \Sigma_u \neq \emptyset$, then for all $\sigma \in Pos(q) \cap \Sigma_c$,

$$\frac{p_2(r, \sigma)}{p_1(q, \sigma)} \sum_{\alpha \in \Sigma_u} p_1(q, \alpha) + \sum_{\alpha \in Pos(q) \cap \Sigma_c} p_2(r, \alpha) \leq 1.$$

Conditions (i) and (ii) together are necessary and sufficient for the existence of a probabilistic supervisor solving the PSCP. The first part of both conditions corresponds to controllability as used in classical supervisory theory (namely, the condition $Pos(q) \cap \Sigma_u = Pos(r) \cap \Sigma_u$ of (i), and $Pos(r) \cap \Sigma_c \subseteq Pos(q) \cap \Sigma_c$ of (ii)). The remaining equations and inequalities correspond to the conditions for probability matching.

Theorem 2: Assume that the conditions (i) and (ii) of Theorem 1 are satisfied. Let $\Gamma = Pos(q) \cap \Sigma_c$ if $Pos(q) \cap \Sigma_u \neq \emptyset$, and $\Gamma = (Pos(q) \cap \Sigma_c) \backslash \{\gamma\}$ otherwise, where $\gamma \in Pos(q)$ is such that for every $\sigma \in Pos(q)$, $\frac{p_2(r,\gamma)}{p_1(q,\gamma)} \geq \frac{p_2(r,\sigma)}{p_1(q,\sigma)}$ is satisfied. Let $x_s^0 \in [0,1]^\Gamma$ and $f : \mathbb{R}^\Gamma \to \mathbb{R}^\Gamma$. For $x_s^0 = 0$, the sequence

$$x_s^{k+1} = f(x_s^k), \qquad k = 0, 1, \ldots, \text{ where}$$

$$f_\sigma(x_s) = \frac{p_2(r,\sigma)}{p_1(q,\sigma) h_\sigma(x_s)}, \sigma \in \Gamma \text{ where}$$

$$h_\sigma(x_s) = \sum_{\Theta \in \mathcal{P}(\Gamma \backslash \{\sigma\})} \frac{1}{1 - \sum_{\alpha \in \Theta} p_1(q,\alpha)} \prod_{\alpha \in \Theta} (1 - x_{s,\alpha}) \prod_{\alpha \in \Gamma \backslash \{\sigma\} \backslash \Theta} x_{s,\alpha}$$

converges to the control input x_s^* (i.e., $V_p(s,\sigma) = x_{s,\sigma}^*$ for $\sigma \in \Gamma$).

The following results are for prefix closed probabilistic specification languages so in the sequel we simplify the probabilistic generator $G = (Q, \Sigma, \delta_1, q_0, Q_m, p)$ to $G = (Q, \Sigma, \delta_1, q_0, p)$.

III. The metric for the closest approximation

In this section, we roughly describe the problem of optimal supervisory control, introduce research done on metrics on states of probabilistic systems, and present the chosen metric.

A. Formulation of the Problem

In the case when the conditions for the existence of a solution to the probabilistic supervisory control problem are not satisfied, we search for a suitable approximation. Assume that the plant is given as $G_1 = (Q, \Sigma, \delta_1, q_0, p_1)$ and the specification is $G_2 = (R, \Sigma, \delta_2, r_0, p_2)$. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$, we seek V_p such that $L_p(V_p/G_1) = L_p(G_3)$, where $G_3 = (S, \Sigma, \delta_1, s_0, p_3)$ and G_3 is closest to G_2 according to a certain criterion. As a criterion, we choose a *pseudometric* on the initial states of generators as explained in the sequel.

B. Literature review

Probabilistic bisimulation is commonly used to define an equivalence relation between probabilistic systems. However, probabilistic bisimulation is hardly a robust relation: roughly speaking, two states of probabilistic systems are bisimilar if and only if they have the same transitions with exactly the same probabilities to states in the same equivalence classes. The formal definition follows and represents a modified version of the definition of bisimulation given in [19].

Definition 1: Let $G = (Q, \Sigma, \delta, q_0, p)$ be a PDES. Probabilistic bisimulation on Q is the binary relation $\equiv$ such that for any $q_1 \equiv q_2$ and $\sigma \in \Sigma$, the following holds:

1) For every q_1' such that $\delta(q_1, \sigma) = q_1'$, there is q_2' such that $\delta(q_2, \sigma) = q_2'$, $p_1(q_1, \sigma) = p_1(q_2, \sigma)$, and $q_1' \equiv q_2'$.
2) For every q_2' such that $\delta(q_2, \sigma) = q_2'$, there is q_1' such that $\delta(q_1, \sigma) = q_1'$, $p_1(q_1, \sigma) = p_1(q_2, \sigma)$, and $q_1' \equiv q_2'$.

States q_1 and q_2 are probabilistic bisimilar if there exists a probabilistic bisimulation $\equiv$ such that $q_1 \equiv q_2$.

Note that we did not use the definition of probabilistic bisimulation for reactive systems as given in [20] or the definition of probabilistic bisimulation for generative systems given in [21]. Since our model is deterministic, we found the straightforward modification of [19] sufficient for our needs.

As a more flexible way to compare probabilistic systems, a notion of *pseudometric* is introduced. A *pseudometric on a set of states* Q is a function $d : Q \times Q \to \mathbb{R}$ that defines a distance between two elements of Q, and satisfies the following conditions: $d(x, y) \geq 0$, $d(x, x) = 0$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$, for any $x, y, z \in Q$. If all distances are not greater than 1, the pseudometric is *1-bounded*. In the sequel, we will use the terms metric and pseudometric interchangeably.

The first paper that discussed the use of a metric as a way to measure the distance between two probabilistic processes is [22]. This early work considers deterministic probabilistic systems. The distance between processes is a number between 0 and 1, and represents a measure of a behavioral proximity between the processes: the smaller the number, the smaller the distance.

Labeled Markov chains are considered in [23]. The motivation of the paper is to explore the possibility of substituting one process with another that is sufficiently close in a metric space. The pseudometric is given via a real-valued logic that is motivated by the well-known result that the Hennessy-Milner logic is complete for bisimulation [24]. Further, this metric is inspired by the Kantorovich metric [25] which is used in transport problems, and more recently has been used by Hutchinson in his theory of fractals [26]. Two states are at distance 0 in this metric if and only if they are probabilistic bisimilar.

More concretely, the ideas of [27] are used to generalize logic so that reasoning about probabilistic systems is supported. A function $f \in \mathcal{F}$ evaluated at a state takes truth values in the interval $[0, 1]$, instead of $\{0, 1\}$. Then, the distance between two states is defined as a pseudometric:

$$d(q_q, q_r) = sup\{|f(q_q) - f(q_r)| f \in \mathcal{F}\}.$$

This paper also offers an algorithm to calculate a distance between two systems in the introduced metric with a prespecified accuracy. The algorithm runs in exponential time. Further, asymptotic metrics have been considered as a side topic. An extension of [23] is given in [28]. It considers not only discrete probabilistic systems (labeled Markov chains), but also continuous systems (labeled Markov processes). In [29] only continuous-time labeled Markov processes are considered. The paper also shows that a small perturbation in a system events' probabilities results in a system that is close to the original one. A pseudometric analogue to weak bisimulation is presented in [30] for labeled concurrent Markov chains where the set of states is divided into two disjoint sets of probabilistic and nondeterministic states. The transitions from probabilistic states are not labeled, while the transitions from nondeterministic states are. The metric is given a fixed-point characterization that allows for coinductive definition.

The systems considered in [31] are reactive nonlabeled probabilistic systems, but the results can be easily generalized to labelled systems [32]. The paper suggests a pseudometric as a measure of behavioral similarity between the systems similar to the one presented in [23]. The pseudometric is coalgebraic as opposed to that of [23]. The paper further presents an algorithm to calculate the distance between two systems in the presented pseudometric with a prespecified accuracy. Comparison of this metric with a number of metrics (e.g., [33], [34]) is given in [35]. Also, the metric can be recovered from the metric of [23] by adding negation to the logic of [23] (see [35]).

The work of [17] is closely related to [23], [30], [31]. It introduces a pseudometric on states for a large class of probabilistic automata, including reactive and generative probabilistic automata. The pseudometric is based on Kantorovich metric on distributions and is characterized as the greatest point of a function. The metric to be used in the solution of our problem is based on this metric. It intuitively matches our notion of the distance between PDESs, and accounts for all differences between corresponding transition probabilities, as opposed to e.g., that of [22] that, roughly speaking, considers only the maximum of the differences between the corresponding probabilities. Furthermore, as the metric is suggested for a large class of systems, it allows for an extension of our work to e.g., nondeterministic systems. Also, as it turns out, there is a simple algorithm to compute distances in this metric for our generative, deterministic model.

C. The metric

Let $G = (Q, \Sigma, \delta, q_0, p)$ be a nonterminating PDES, where $Q = \{q_0, q_1, \ldots q_{N-1}\}$ (if a plant is terminating, it can easily be transformed into a nonterminating one using the technique described in [9]). This is the system we shall be using throughout the sequel. Next, we introduce some useful notation. Let $q_q, q_r \in Q$ and let ρ_{q_q} and ρ_{q_r} be the distributions on $\Sigma \times Q$ induced by the states q_q and q_r, respectively. Assume $0 \leq i, j \leq N-1$ and $\Psi = Pos(q_q) \cup Pos(q_r)$. For notational convenience, we will write $\rho_{\sigma,i}$ instead of $\rho_{q_q}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_{q_r}(\sigma, q_j)$.

Next, we present the pseudometric suggested in [17] for a large class of automata which includes our generator. This pseudometric is also based on Kantorovich metric on probability measures.

First, [17] introduces the class $\mathcal{M}$ of 1-bounded pseudometrics on states with the ordering

$$d_1 \preceq d_2 \text{ if } \forall s, t \; d_1(s,t) \geq d_2(s,t) \tag{1}$$

as was initially suggested in [30]. It is proved that $(\mathcal{M}, \preceq)$ is a complete lattice.

Then, let $d \in \mathcal{M}$, and let the constant $e \in (0, 1]$ be a *discount factor* that determines the degree to which the difference in the probabilities of farther transitions is discounted: the smaller the value of e, the greater the discount on future transitions. We assume that the total mass of ρ_{q_q} is greater or equal to the total mass of ρ_{q_r}:

$$\sum_{\substack{\sigma \in \Psi \\ 0 \leq i \leq N-1}} \rho_{\sigma,i} \geq \sum_{\substack{\sigma \in \Psi \\ 0 \leq i \leq N-1}} \rho'_{\sigma,i}.$$

This assumption is not needed for nonterminating automata. Then, the distance between the distributions ρ_{q_q} and ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$ (note a slight abuse of notation) for our generators is given as:

$$\begin{aligned}
&\text{Maximize} \quad && e \cdot \left(\sum_{\substack{\sigma \in \Psi \\ 0 \le i \le N-1}} a_{\sigma,i}\rho_{\sigma,i} - \sum_{\substack{\sigma \in \Psi \\ 0 \le i \le N-1}} a_{\sigma,i}\rho'_{\sigma,i} \right) \\
&\text{subject to} \quad && 0 \le a_{\sigma,i} \le 1, \qquad \sigma \in \Psi,\ \ 0 \le i \le N-1 \\
& && a_{\sigma,i} - a_{\alpha,j} \le c_{ij}^{\sigma\alpha}, \quad \sigma, \alpha \in \Psi,\ \ 0 \le i, j \le N-1 \qquad (2) \\
& && \text{where} \\
& && c_{ij}^{\sigma\alpha} = \begin{cases} d(q_i, q_j) & \text{if } \sigma = \alpha \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

If the total mass of ρ_{q_q} is less than the total mass of ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$ is defined to be $d(\rho_{q_r}, \rho_{q_q})$. This extension to distributions is also pseudometric, and is consistent with the ordering (1) (see [30]).

The pseudometric on states, d_d, is, then, given as the greatest fixed-point of the function $\mathcal{D}$ on $\mathcal{M}$ (here we give a simplified version of that in [17]):

$$\mathcal{D}(d)(q_q, q_r) = d(\rho_{q_q}, \rho_{q_r}), \quad d \in \mathcal{M}, q_q, q_r \in Q \qquad (3)$$

The proofs that the function defined by (3) is monotone on $\mathcal{M}$, that it does have the greatest fixed point, and that its closure ordinal is ω, originate from [30]. Furthermore, since the pseudometric d_d is a fixed-point of (3), the distances in d_d are between 0 and e. In order to scale the distances with the factor $1/e$ so that they can be between 0 and 1, we define a pseudometric d_m for nonterminating generators to be the greatest fixed-point of the function (3) on $\mathcal{M}$, where

$d(\rho_{q_q}, \rho_{q_r})$ is redefined to be:

$$\begin{array}{lll}
\text{Maximize} & \sum\limits_{\substack{\sigma\in\Psi\\ 0\le i\le N-1}} a_{\sigma,i}\rho_{\sigma,i} - \sum\limits_{\substack{\sigma\in\Psi\\ 0\le i\le N-1}} a_{\sigma,i}\rho'_{\sigma,i} & \\
\text{subject to} & 0 \le a_{\sigma,i} \le 1, \qquad \sigma \in \Psi,\ \ 0 \le i \le N-1 & \\
& a_{\sigma,i} - a_{\alpha,j} \le c_{ij}^{\sigma\alpha}, \quad \sigma, \alpha \in \Psi,\ \ 0 \le i, j \le N-1 & \\
& \text{where} & \\
& c_{ij}^{\sigma\alpha} = \begin{cases} e \cdot d(q_i, q_j) & \text{if } \sigma = \alpha \\ 1 & \text{otherwise} \end{cases} &
\end{array} \tag{4}$$

Given the aforementioned difference between our definition of the distance between distributions and that of [17], it is obvious that the the distances between states in our pseudometric are by the factor $1/e$ larger than theirs.

At this point we recall the fact that our generators are deterministic: for an event σ and a state q, there is at most one state q' such that $q' = \delta(q, \sigma)$. For the purposes of the following analysis of our nonterminating deterministic generators, we rewrite the objective function of the optimization problem of (4) as:

$$\sum_{\sigma\in\Psi} \left(a_{\sigma,i(q_q,\sigma)}\rho_{\sigma,i(q_q,\sigma)} - a_{\sigma,j(q_r,\sigma)}\rho'_{\sigma,j(q_r,\sigma)}\right) \tag{5}$$

where $i(q_q, \sigma) = i$ such that $q_i = \delta(q_q, \sigma)$ if $\delta(q_q, \sigma)!$, and $i(q_q, \sigma) = 0$, otherwise. We arbitrarily choose $i(q_q, \sigma)$ to be 0 when $\delta(q_q, \sigma)$ is not defined although we could have chosen any other $i \in \{1, \dots, N-1\}$. This is because when $\delta(q_q, \sigma)!$ does not hold, then $\rho_{\sigma,i(q_q,\sigma)} = 0$ for any $i(q_q, \sigma) \in \{1, \dots, N-1\}$. Similarly, $j(q_r, \sigma) = j$ such that $q_j = \delta(q_r, \sigma)$ if $\delta(q_r, \sigma)!$, and $j(q_r, \sigma) = 0$, otherwise. For readability purposes, we will write i instead of $i(q_q, \sigma)$, and j instead of $j(q_r, \sigma)$.

Also, since our generators are nonterminating, we do not need the first constraint in (4); we keep it, however, as it will be useful later. Further, the objective function of this linear programming problem can be maximized by maximizing each of its summands separately. In order to explain this observation, we consider a summand $a_{\sigma,i}\rho_{\sigma,i} - a_{\sigma,j}\rho'_{\sigma,j}$. Due to the generator's determinism, there is no other nonzero summand containing $a_{\sigma,k}$, $0 \le k \le N-1$, $k \ne i$, $k \ne j$.

Therefore, the last constraint of (4) for any two coefficients $a_{\sigma,i}$ and $a_{\alpha,j}$ ($0 \leq i, j \leq N-1$) from different summands becomes $a_{\sigma,i} - a_{\alpha,j} \leq 1$. This constraint is already implied by the first constraint, so we can independently pick the coefficients a in different summands, and, consequently, independently maximize the summands in order to maximize the sum.

In order to maximize a summand of the objective function (5), we solve the following linear programming problem for $\sigma \in \Psi$:

$$\begin{aligned} \text{Maximize} \quad & (a_{\sigma,i}\rho_{\sigma,i} - a_{\sigma,j}\rho'_{\sigma,j}) \\ \text{subject to} \quad & 0 \leq a_{\sigma,i}, a_{\sigma,j} \leq 1, \\ & a_{\sigma,i} - a_{\sigma,j} \leq c_{ij} \end{aligned} \tag{6}$$

where i and j are defined as in (5), and $c_{ij} = ed(q_i, q_j)$ as before. Also, note that the set of constraints does not contain the inequality $a_{\sigma,j} - a_{\sigma,i} \leq c_{ji}$. In order to maximize the given function, the coefficient $a_{\sigma,i}$ is to be chosen to be greater than $a_{\sigma,j}$ since the given constraints allow it. In that case, since $c_{ij} = c_{ji}$, if $a_{\sigma,i} - a_{\sigma,j} \leq c_{ij}$, then $a_{\sigma,j} - a_{\sigma,i} \leq c_{ji}$ follows, so the latter constraint is redundant. Further, it is not hard to see that the solution of the given linear programming problem for $\rho_{\sigma,i} \geq \rho'_{\sigma,j}$ is equal to $\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$. We can solve this problem using graphical method, simplex method or using the following line of reasoning. In order to maximize the given function, we can either choose $a_{\sigma,i}$ to be 1 and then pick $a_{\sigma,j}$ so that it has the minimal value for the given constraints, or we choose $a_{\sigma,j}$ to be 0, and then pick $a_{\sigma,i}$ so that it has the maximal value under the given constraints. In the first case, we pick $a_{\sigma,i}$ to be 1, $a_{\sigma,j}$ to be $1 - c_{ij}$, and value of the objective function is $\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$. In the second case, since $a_{\sigma,j}$ is 0, then $a_{\sigma,i}$ is equal to c_{ij}, and the objective function becomes $c_{ij}\rho_{\sigma,i}$. Since $c_{ij}\rho_{\sigma,i}$ is never greater than $\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$ (for $\rho_{\sigma,i} \geq \rho'_{\sigma,j}$ and $c_{ij} \in [0, 1]$), the latter is our solution. Using the same reasoning, for $\rho_{\sigma,i} < \rho'_{\sigma,j}$, the maximum is reached at $(a_i, a_j) = (c_{ij}, 0)$ and its value is $c_{ij}\rho_{\sigma,i}$.

Now, we put together the presented solution of the linear programming problem (4) and the aforementioned reasoning. The distance between the distributions ρ_{q_q} and ρ_{q_r} is then:

$$d(\rho_{q_q}, \rho_{q_r}) = \sum_{\sigma \in \Psi} f(q_q, q_r, d, \sigma), \text{ where}$$

$$f(q_q, q_r, d, \sigma) = \begin{cases} \rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j} & \text{if } \rho_{\sigma,i} \geq \rho'_{\sigma,j} \\ c_{ij}\rho_{\sigma,i} & \text{otherwise} \end{cases} \tag{7}$$

$$or, equivalently,$$

$$f(q_q, q_r, d, \sigma) = max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}),$$

where $c_{ij} = e \cdot d(q_i, q_j)$ as before, and i and j denote $i(q_q, \sigma)$ and $j(q_r, \sigma)$, respectively, as defined as in (5).

To summarize, the function $\mathcal{D}(d)$ for our model is given as:

$$\mathcal{D}(d)(q_q, q_r) = \sum_{\sigma \in \Psi} max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}) \tag{8}$$

where, again, $c_{ij} = e \cdot d(q_i, q_j)$ as before, and i and j denote $i(q_q, \sigma)$ and $j(q_r, \sigma)$, respectively, as defined in (5).

D. Calculating the Pseudometric: Algorithms

For $e \in (0, 1)$, we will prove that, for our model, the function $\mathcal{D}$ as given in (8) has only one fixed point. Then, we suggest two algorithms for calculating the distances in the pseudometric that represents the fixed point of this function.

First, we introduce some useful definitions and results from linear algebra.

A real $n \times n$ matrix $A = (a_{ij})$ defines a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$, and we will write $A \in L(R^n)$ to denote either the matrix or linear function, as we shall make no distinction between the two.

Definition 2: For any complex $n \times n$ matrix A, the spectral radius of A is defined as the maximum of $|\lambda_1|, \cdots, |\lambda_n|$, where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of A.

The spectral radius of A, denoted $\varphi(A)$, satisfies $\varphi(A) \leq ||A||$, where $||A||$ is an arbitrary norm on $\mathbb{R}^n$. During the course of the following proof, we will make use of infinity norm $||A||_\infty = \underset{1 \leq i \leq n}{max} \sum_{j=1}^{n} |a_{ij}|$.

Definition 3: An operator $G : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is called a *P-contraction* on a set $D_0 \subseteq D$ if there exists a linear operator $P \in L(\mathbb{R}^n)$ such that $P \geq 0$, $\varphi(P) < 1$ and

$$|G(x) - G(y)| \leq P\,|x - y| \text{ for all } x, y \in D_0 \tag{9}$$

Now, let $d \in \mathcal{M}$. Next, we define the function $\mathcal{V}$ on $\mathcal{M}$:

$$\mathcal{V}(d) = (d(q_0, q_0), d(q_0, q_1), \ldots, d(q_{N-1}, q_{N-1}))^T.$$

Therefore, $\mathcal{V}(d) = (\mathcal{V}_1(d), \mathcal{V}_2(d), \cdots, \mathcal{V}_K(d))^T$, where $\mathcal{V}_k(d)$ for $k \in \{1, \ldots, K\}$ is given as:

$$\mathcal{V}_k(d) = d(q_i, q_j),\ i = k\ div\ (N+1),\ j = (k-1)\ mod\ N.$$

Now, we extend the function $\mathcal{D}$ in a natural way to the function $\mathcal{D}(\mathcal{V}(d)) = (\mathcal{D}_1(\mathcal{V}(d)), \ldots, \mathcal{D}_K(\mathcal{V}(d)))^T$, where for any $k \in \{1, \ldots, K\}$:

$$\mathcal{D}_k(\mathcal{V}(d)) = d(\rho_{q_i}, \rho_{q_j}),\ \ i = k\ div\ (N+1),\ j = (k-1)\ mod\ N. \tag{10}$$

Further, let $D_0 = \{\mathcal{V}(d) | d \in \mathcal{M}\}$.

Lemma 1: The function $\mathcal{D}$ is a P-contraction on D_0.

Proof: Let $d', d'' \in \mathcal{M}$, and $\mathfrak{d}' = \mathcal{V}(d')$, and $\mathfrak{d}'' = \mathcal{V}(d'')$. Let $k \in \{1, \ldots, K\}$, and let i and j ($0 \leq i, j \leq N-1$) be given as in (10). Further, assume $\Psi = Pos(q_i) \cup Pos(q_j)$. Also, let $t(i, \sigma) = t$ such that $\delta(q_i, \sigma) = q_t$ if $\delta(q_i, \sigma)!$, and $t(i, \sigma) = 0$, otherwise. Similarly, $l(j, \sigma) = l$ such that $\delta(q_j, \sigma) = q_l$ if $\delta(q_j, \sigma)!$, and $l(j, \sigma) = 0$, otherwise. Again, for notational convenience, we will write t instead of $t(i, \sigma)$, and l instead of $l(j, \sigma)$. Also, we will write $\rho_{\sigma,t}$ instead of $\rho_{q_i}(\sigma, q_t)$, and, similarly, $\rho'_{\sigma,l}$ instead of $\rho_{q_j}(\sigma, q_l)$ for $q_t, q_l \in Q$. Then:

$$\begin{aligned}
&|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| = \left|d'(\rho_{q_i}, \rho_{q_j}) - d''(\rho_{q_i}, \rho_{q_j})\right| \\
&= \left|\sum_{\sigma \in \Psi} max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed'(q_t, q_l)\rho'_{\sigma,l}, ed'(q_t, q_l)\rho_{\sigma,t}) - \sum_{\sigma \in \Psi} max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed''(q_t, q_l)\rho'_{\sigma,l}, ed''(q_t, q_l)\rho_{\sigma,t})\right| \\
&= \left|\sum_{\sigma \in \Psi} e(d'(q_t, q_l) - d''(q_t, q_l)) min(\rho_{\sigma,t}, \rho'_{\sigma,l})\right| \\
&\leq \sum_{\sigma \in \Psi} emin(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |d'(q_t, q_l) - d''(q_t, q_l)| &(11)\\
&\leq \sum_{\substack{\sigma \in \Psi \\ m = tN+l+1}} emin(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |\mathfrak{d}'_m - \mathfrak{d}''_m|\,. &(12)
\end{aligned}$$

Note that as $t = t(i, \sigma)$ and $l = l(j, \sigma)$ are also functions of k (since i and j are functions of k). Now, without the explicit construction of the matrix P, we can see from (12) that there exists

P such that $|\mathcal{D}(\mathfrak{d}') - \mathcal{D}(\mathfrak{d}'')| \leq P\,|\mathfrak{d}' - \mathfrak{d}''|$ where

$$\begin{aligned}
||P||_\infty &= \max_k \sum_{\sigma\in\Psi} emin(\rho_{\sigma,t}, \rho'_{\sigma,l}) \\
&\leq e \\
&\text{(since } \sum_{\substack{\sigma\in\Psi \\ t\in\{0,\ldots,N-1\}}} \rho_{\sigma,t} = \sum_{\substack{\sigma\in\Psi \\ l\in\{0,\ldots,N-1\}}} \rho'_{\sigma,l} = 1) \\
&< 1 \text{ (since } e \in (0,1))
\end{aligned}$$

Therefore, $\varphi(P) < 1$ and, since, obviously, $P \geq 0$, then $\mathcal{D}$ is P-contraction. ∎

Lemma 2: Let $d', d'' \in \mathcal{M}$, and $\mathfrak{d}' = \mathfrak{d}(d')$, and $\mathfrak{d}'' = \mathfrak{d}(d'')$. For any $k \in \{1, \ldots, K\}$, there exists $m \in \{1, \ldots, K\}$ such that:

$$|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| \leq e\,|\mathfrak{d}'_m - \mathfrak{d}''_m|$$

Proof: We use the notation from the previous lemma.

$$\begin{aligned}
&|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| \\
&\leq \sum_{\sigma\in\Psi} emin(\rho_{\sigma,t}, \rho'_{\sigma,l})\,|d'(q_t, q_l) - d''(q_t, q_l)| \text{ (from (11))} \\
&\leq \sum_{\sigma\in\Psi} emin(\rho_{\sigma,t}, \rho'_{\sigma,l}) \max_{\substack{(t,l)\in \\ \{(t(\sigma,i), l(\sigma,j))|\sigma\in\Psi\}}} \{|d'(q_t, q_l) - d''(q_t, q_l)|\} \\
&\leq \sum_{\sigma\in\Psi} emin(\rho_{\sigma,t}, \rho'_{\sigma,l})\,|d'(q_r, q_s) - d''(q_r, q_s)| \quad \text{for some } r, s \in \{0, \ldots, N-1\} \\
&\leq e|d'(q_r, q_s) - d''(q_r, q_s)| \text{ (since } \sum_{\substack{\sigma\in\Psi \\ t\in\{0,\ldots,N-1\}}} \rho_{\sigma,t} = \sum_{\substack{\sigma\in\Psi \\ l\in\{0,\ldots,N-1\}}} \rho'_{\sigma,l} = 1) \\
&\leq e\,|\mathfrak{d}'_m - \mathfrak{d}''_m| \text{ for some } m \in \{1, \ldots, K\}
\end{aligned}$$

∎

Theorem 3: For any $\mathfrak{d}^0 \in D_0$, the sequence

$$\mathfrak{d}^{n+1} = \mathcal{D}(\mathfrak{d}^n), \;\; n = 0, 1, \ldots$$

converges to the only fixed point of $\mathcal{D}$ in D_0, $\mathfrak{d}^*$, and the error of convergence is given componentwise ($k \in \{1, \ldots, K\}$) as:

$$|\mathfrak{d}^n_k - \mathfrak{d}^*_k| \leq (1-e)^{-1} e^n, \;\; n = 1, 2, \ldots \tag{13}$$

Proof: Note that this is a variant of the contraction-mapping theorem extended to P-contractions ([36], Theorem 13.1.2.). We employ a similar proof technique. Let $n, m \geq 1$. Then:

$$
\begin{aligned}
&|\mathfrak{d}_k^{n+m} - \mathfrak{d}_k^n| \leq \sum_{t=1}^{m} |\mathfrak{d}_k^{n+t} - \mathfrak{d}_k^{n+t-1}| \\
&\leq \sum_{t=1}^{m} e^t |\mathfrak{d}_{i(t)}^n - \mathfrak{d}_{i(t)}^{n-1}| \text{ (applying Lemma 2 } t \text{ times, } i(t) \in \{1, \ldots, K\}) \\
&\leq \sum_{t=1}^{m} e^t \underset{i(t)}{max} \{|\mathfrak{d}_{i(t)}^n - \mathfrak{d}_{i(t)}^{n-1}|\} \\
&\leq \left(\sum_{t=1}^{m} e^t\right) |\mathfrak{d}_j^n - \mathfrak{d}_j^{n-1}| \text{ (for some } j \in \{1, \ldots, K\}) \\
&\leq (1-e)^{-1}\, e |\mathfrak{d}_j^n - \mathfrak{d}_j^{n-1}| \text{ (since } \sum_{t=0}^{m} e^t \leq (1-e)^{-1} \text{ for } m \geq 0) \\
&\leq (1-e)^{-1} e^n |\mathfrak{d}_l^1 - \mathfrak{d}_l^0| \text{ (for some } l\text{, by applying Lemma 2 } (n-1) \text{ times)} \\
&\leq (1-e)^{-1} e^n
\end{aligned}
\tag{14}
$$

Therefore, the sequence $\{\mathfrak{d}_k^n\}_{n \geq 0}$ is a Cauchy sequence and hence converges to some $\mathfrak{d}_k^*$, and, consequently, the sequence $\{\mathfrak{d}^n\}_{n \geq 0}$ converges to $\mathfrak{d}^* \in D_0$. Also, we have:

$$
|\mathfrak{d}^* - \mathcal{D}(\mathfrak{d}^*)| \leq |\mathfrak{d}^* - \mathfrak{d}^{n+1}| + |\mathcal{D}(\mathfrak{d}^n) - \mathcal{D}(\mathfrak{d}^*)| \leq |\mathfrak{d}^* - \mathfrak{d}^{n+1}| + P|\mathfrak{d}^n - \mathfrak{d}^*|
$$

When we let $n \to \infty$, we see that $\mathfrak{d}^* = \mathcal{D}(\mathfrak{d}^*)$. Also, if we let $n \to \infty$ in (14), we get the componentwise error estimate of (13).

At last, we prove that $\mathfrak{d}^*$ is the only fixed point in D_0. Assume that there is another fixed point of $\mathcal{D}$ in the same interval D_0, $\mathfrak{d}^+$. Then,

$$
|\mathfrak{d}^* - \mathfrak{d}^+| = |\mathcal{D}(\mathfrak{d}^*) - \mathcal{D}(\mathfrak{d}^+)| \leq P|\mathfrak{d}^* - \mathfrak{d}^+|
$$

Hence, $(I - P)|\mathfrak{d}^* - \mathfrak{d}^+| \leq 0$. However, since $\varphi(P) < 1$, $(I - P)^{-1} = \sum_{i=0}^{\infty} P^i \geq 0$, then $|\mathfrak{d}^* - \mathfrak{d}^+| \leq 0$. Therefore, $\mathfrak{d}^* = \mathfrak{d}^+$. ∎

Now, using the presented analysis, we suggest the following two algorithms for the calculation of the distances between the states of PDESs in the chosen pseudometric.

Algorithm 1 Theorem 3 proves that the system of equations

$$
\mathfrak{d} = \mathcal{D}(\mathfrak{d}) \tag{15}
$$

has a unique solution. The equations are linear. Therefore, the system (15) can be rewritten into the standard form $A\mathfrak{d} = b$, where A is a $K \times K$ matrix and b is a column vector of dimension K. Therefore, the distances in our pseudometric can be calculated by solving this system of linear equations. The distances found are exact solutions (if we disregard the round-off error).

Algorithm 2 Theorem 3 suggests an iterative algorithm to calculate distances between the states of a probabilistic generator. The algorithm is a straightforward modification of that of [31] that calculates distances in a pseudometric suggested for a different kind of probabilistic system and derived by using terminal coalgebras. Let $d^0(q_q, q_r) = 0$ for any two states $q_q, q_r \in Q$. As before, let ρ_{q_q} and ρ_{q_r} be the distributions induced by the states q_q and q_r, respectively. Assume $0 \le i, j \le N-1$ and $\Psi = Pos(q_q) \cup Pos(q_r)$. The n-th iteration of the algorithm calculates the distance d^n between each two states $q_q, q_r \in Q$:

$$d^n(q_q, q_r) = \sum_{\sigma \in \Psi} max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}) \tag{16}$$

where $c_{ij} = e \cdot d^{n-1}(q_i, q_j)$, and $i = i(q_q, \sigma)$ and $j = j(q_r, \sigma)$ are defined as in (5). The accuracy of the solution found in n-th iteration is $(1-e)^{-1}e^n$.

The iterative method can be useful for systems with large K, where the direct method can be rather expensive. Furthermore, the mathematical apparatus used to reach the iterative method will be reused in the solution of the closest approximation problem.

IV. Finding the closest approximation

In this section we first characterize the closest approximation and then give the algorithm to minimize the distance between the required behaviour of a system and its achievable behaviour. All the results in the sequel are applicable for $e \in (0, 1)$.

A. Characterizing The Closest Approximation

First, we repeat the formulation of the nearest approximation problem. Assume that the plant is given as PDES $G_p = (Q_p, \Sigma, \delta_p, q_{p_0}, p_p)$, and the requirements specification is given as $G_r = (Q_r, \Sigma, \delta_r, q_{r_0}, p_r)$. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$, we seek the optimal (closest) solution and characterize it as follows. As mentioned before, the conditions (i) and (ii) of the Theorem 1 for the existence of probabilistic supervisor consist of two parts. The first part of both conditions corresponds to controllability as used in classical supervisory theory (namely, the condition $Pos(q) \cap \Sigma_u = Pos(r) \cap \Sigma_u$ of (i), and $Pos(r) \cap \Sigma_c \subseteq Pos(q) \cap \Sigma_c$ of (ii)). The remaining equations and inequalities correspond to the conditions for probability matching. Hence, before we start looking for the closest approximation in the sense of probability matching, we resort to the classical supervisory theory of supremal controllable languages. We first find $L(G_p) \cap L(G_r)$, and then the supremal controllable sublanguage of $L(G_p) \cap L(G_r)$ (with respect to G_p), K. Then, the DES that represents this language K, further equipped with p_p distribution (appropriately normalized) becomes the modified plant PDES G_1, and the same DES (corresponding to the supremal controllable language K) equipped with the distribution p_r appropriately normalized becomes the desired behaviour PDES G_2. Formally, let (reachable and deadlock-free) DES $G = (Q, \Sigma, \delta, q_0)$ represent the supremal controllable language K. We define a PDES $G_1 = (Q, \Sigma, \delta, q_0, p)$, where the distribution $p : Q \times \Sigma \rightarrow [0, 1]$, for any $q \in Q, \sigma \in \Sigma$, is defined as:

$$p(q, \sigma) = \frac{p_p(q_p, \sigma)}{\sum\limits_{\sigma \in \{\sigma \in \Sigma | \delta(q,\sigma)!\}} p_p(q_p, \sigma)} \tag{17}$$

where $q_p = \delta_p(q_{p_0}, s)$ for $s \in L(G)$ such that $q = \delta(q_0, s)$. Similarly, we define a PDES $G_2 = (Q, \Sigma, \delta, q_0, p_2)$ where the distribution $p_2 : Q \times \Sigma \rightarrow [0, 1]$, where, for any $q \in Q, \sigma \in \Sigma$:

$$p_2(q, \sigma) = \frac{p_r(q_r, \sigma)}{\sum\limits_{\sigma \in \{\sigma \in \Sigma | \delta(q,\sigma)!\}} p_r(q_r, \sigma)} \tag{18}$$

where $q_r = \delta_r(q_{r_0}, s)$ for $s \in L(G)$ such that $q = \delta(q_0, s)$.

Next, we check probability matching equations and inequalities from Theorem 1. If they are not satisfied (there is no probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$), we seek $G_2' = (Q', \Sigma, \delta', q_0', p')$ such that there exists a probabilistic supervisor V_p so that $L_p(V_p/G_p) = L_p(G_2')$ holds, and G_2' is closest to G_2 in the metric presented. Without loss of generality, we assume that nonprobabilistic automata underlying G_2 and G_2' are isomorphic (with labeling of events being preserved). Therefore, nonprobabilistic automata underlying G_2 and G_2' are identical up to renaming of states. This assumption is not restrictive as there cannot be any string in the desired system that does not belong to $L(G_2)$. Our goal is to find the probabilities of G_2' as the closest approximation such that the distance between G_2 and G_2' (i.e. the distance between the initial states of G_2 and G_2') is minimized. As our metric is defined on the states of a system, in order to define distances between the states of G_2 and G'_2, we can consider the union PDES $G_u = (Q \cup Q', \Sigma, \delta_u, q_0, p_u)$, where

$$\delta_u(q,\sigma) = \delta(q,\sigma), p_u(q,\sigma) = p(q,\sigma) \text{ if } q \in Q \text{ and } \sigma \in \Sigma$$

$$\delta_u(q,\sigma) = \delta'(q,\sigma), p_u(q,\sigma) = p'(q,\sigma) \text{ if } q \in Q' \text{ and } \sigma \in \Sigma$$

Then, $\mathcal{M}$ is the set of 1-bounded pseudometrics on the states of this joined system with the same ordering as in (1).

Further, note that nonprobabilistic automata underlying G_1 and G_2 are also isomorphic (with the isomorphism function being the identity function). We first note that, considering the isomorphism between the nonprobabilistic versions of G_2 and G'_2, we will be interested only in the distances between (probability measures on) states $q \in Q$ of G_2 and $q' = f(q) \in Q'$ of G'_2, where f is the isomorphism between nonprobabilistic automata underlying G_2 and G'_2 (and between nonprobabilistic automata underlying G_1 and G'_2). We assume that, after the occurrence of string $s \in L(G_1)$, the PDES G_1 is in state q ($\delta(q_0, s) = q$). Then, G_2 and its closest approximation G'_2 are in states q and q', respectively, and $q' = f(q)$.

Let us first define a class $\mathcal{A}$ of functions $s : Q \times Q' \to [0,1]$, such that $\forall q \in Q, q' = f(q) \in Q'$ $s(q,q') = d(q,q')$, where $d \in \mathcal{M}$. Therefore, the class $\mathcal{A}$ is the class of all the pseudometrics with domain reduced only to $Q \times Q'$: only distances between $q \in Q$ and $q' = f(q) \in Q'$ are defined. Next, we define a class $\mathcal{A}$ of partial functions $b : Q \times Q' \to [0,1]$, such that $\forall q \in Q, q' = f(q) \in Q'$ $b(q,q') = d(q,q')$, where $d \in \mathcal{M}$. Therefore, the class $\mathcal{A}$ is the class of all 1-bounded pseudometrics with domain reduced to $Q \times Q'$, and only distances between $q \in Q$ and $q' = f(q) \in Q'$ defined. Next, we define a family of functions $\Delta = \{\rho' | \rho'(q') \text{ is a probability distribution on } \Sigma \times Q' \text{ induced by state } q' \in Q'\}$. Now, for each $\rho' \in \Delta$, we define function $\mathcal{D}^{\rho'} : \mathcal{A} \to \mathcal{A}$ as ($q \in Q, q' = f(q) \in Q', d \in \mathcal{A}$):

$$\mathcal{D}^{\rho'}(d)(q,q') = d(\rho_q, \rho'_{q'}) \text{ and } \rho'(q') = \rho'_{q'}$$

where, as before, d is lifted to the metric on distributions, and $d(\rho_q, \rho'_{q'})$ is defined as in (7). Further, for each $\rho' \in \Delta$, we define function $d^{\rho'}_{max}$ as the greatest fixed point of function $\mathcal{D}^{\rho'}$. We introduce the reversed ordering on $\mathcal{A}$ to match the one in (1):

$$d_1 \preceq' d_2 \text{ if } \forall q \in Q q' \in Q' d_1(q,q') \geq d_2(q,q').$$

The fact that $(\mathcal{A}, \preceq')$ is a complete lattice follows from the fact that $(\mathcal{M}, \preceq)$ is a complete lattice. Let $\rho' \in \Delta$. The problem of finding optimal approximation reduces now to finding $\rho'_m \in \Delta$ such that $d^{\rho'_m}_{max}(q_0, q'_0) = \underset{\rho'}{min}\{d^{\rho'}_{max}(q_0, q'_0)|\rho' \in \Delta\}$ and the conditions for the existence of a probabilistic supervisor of Theorem 1 are satisfied. It follows straight from the definitions of $\mathcal{D}^{\rho'}$ and $d^{\rho'}_{max}$ that, for any $\rho' \in \Delta$, $q \in Q, q' = f(q) \in Q'$, the distances $d^{\rho'}_{max}(q, q')$ for $\rho' \in \Delta$ are distances in our pseudometric.

Further, we are now able to substantially simplify the notation used in the previous section. We assume that $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and $Q' = \{q'_0, q'_1, \ldots q'_{N-1}\}$, where $q'_i = f(q_i)$, $i = 0, \ldots, N-1$. Also, assume $\Psi = Pos(q)$, $\Psi_u = Pos(q) \cap \Sigma_u$, and $\Psi_c = Pos(q) \cap \Sigma_c$. For PDES G_2, let ρ_q be the probability distribution induced by the state $q \in Q$ and, for PDES G'_2, let $\rho'_{q'}$ be the probability distribution induced by the state $q' \in Q'$. Also, we will write ρ_σ instead of $\rho_q(\sigma, q_i)$, ρ'_σ instead of $\rho'_{q'}(\sigma, q'_i)$ and p_σ instead of $p(q, \sigma)$. Next we define the function $\mathcal{P} : \mathcal{A} \to \mathcal{A}$ ($q \in Q, q' = f(q) \in Q', d \in \mathcal{A}$):

$$\mathcal{P}(d)(q, q') = \underset{\rho'_\sigma}{\text{Minimize}} \sum_{\sigma \in \Psi} max(\rho_\sigma - \rho'_\sigma + c_i\rho'_\sigma, c_i\rho_\sigma), \tag{19}$$

where $c_i = e \cdot d(q_i, q'_i)$ s.t. $q_i = \delta(q, \sigma)$

subject to

$$\frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha} = \frac{\rho'_\sigma}{\sum\limits_{\alpha \in \Psi_u} \rho'_\alpha}, \qquad \sigma \in \Psi_u, \tag{20}$$

$$\frac{\sum_{\alpha \in \Psi_u} p_\alpha}{p_\sigma}\rho'_\sigma + \sum_{\alpha \in \Psi_c} \rho'_\alpha \leq 1, \quad \sigma \in \Psi_c, \tag{21}$$

$$0 \leq \rho'_\sigma \leq 1, \qquad \sigma \in \Psi, \tag{22}$$

$$\sum_{\alpha \in \Psi} \rho'_\alpha = 1 \tag{23}$$

The function $\mathcal{P}$ is well-defined since, if $\rho'_\sigma = 0$ for $\rho'_\sigma \in \Psi_c$, and if $\rho'_\sigma = \frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha}$ for $\rho'_\sigma \in \Psi_u$, the constraints (20), (21), (22), (23) are satisfied; therefore, the optimization problem has a feasible origin. Since $\mathcal{A}$ is a lattice, and the function $\mathcal{P} : \mathcal{A} \to \mathcal{A}$ is monotone, it has the greatest fixed point.

Next, we prove a lemma.

Lemma 3: Let $(\mathcal{L}, \preceq)$ be a complete lattice, and let $f, g : \mathcal{L} \to \mathcal{L}$ be two monotone functions such that $\forall x \in \mathcal{L} : g(x) \preceq f(x)$. Let $gfp(f)$ and $gfp(g)$ denote the greatest fixed point of functions f and g, respectively. Then, $gfp(g) \preceq gfp(f)$.

Proof: According to Knaster-Tarski theorem, the functions f and g have the greatest fixed points $gfp(f)$ and $gfp(g)$, respectively, where $gfp(f) = sup(\{x|f(x) \preceq x\})$, and $gfp(g) = sup(\{x|g(x) \preceq x\})$. Since $\forall x : g(x) \preceq f(x)$, then $\{x|f(x) \preceq x\} \subseteq \{x|g(x) \preceq x\}$; hence $gfp(g) \preceq gfp(f)$. ∎

Obviously, because of the definition of function $\mathcal{P}$, for any $\rho' \in \Delta$, for function $\mathcal{D}^{\rho'} \in \mathcal{F}$, it holds that $\forall d \in \mathcal{A}\ \mathcal{D}^{\rho'}(d) \preceq' \mathcal{P}(d)$. Using the previous theorem, we conclude that the greatest fixed point of $\mathcal{P}$ is greater than or equal to any $d^{\rho'}_{max} \in \mathcal{A}_{max}$. This greatest fixed point corresponds to the minimal distance between q and q_0 because of the reversed ordering of $\mathcal{A}$. Therefore, the greatest fixed point of function $\mathcal{P}$ corresponds to the pseudometric in which the distance between q_0 and q'_0 is minimized. Consequently, the values of decision variables $\rho'_{q'_i}$ for $q'_i \in Q'$ when the greatest fixed point of $\mathcal{P}$ is reached correspond to the state probability distributions of the optimal approximation.

B. Minimizing the Distance: Algorithm

We suggest an iterative algorithm to calculate minimum achievable distance (i.e. the only fixed point of the function $\mathcal{P}$) up to a desired accuracy and probability distribution of the achievable behaviour of the system when this distance is achieved. We follow the proof pattern used for the algorithm from the Section III-D.

Let $d \in \mathcal{A}$. Again, we assume that $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and $Q' = \{q'_0, q'_1, \ldots q'_{N-1}\}$, where $q'_i = f(q_i)$, $i = 0, \ldots, N-1$. We define the function $\hat{\mathcal{V}}(d) = (\hat{\mathcal{V}}_1(d), \ldots, \hat{\mathcal{V}}_N(d))^T$ as:

$$\hat{\mathcal{V}}(d) = (d(q_0, q'_0), d(q_1, q'_1), \ldots, d(q_{N-1}, q'_{N-1}))^T.$$

Therefore, for $k = 1, \ldots, N$:

$$\hat{\mathcal{V}}_k(d) = d(q_{k-1}, q'_{k-1}). \tag{24}$$

We redefine the function $\mathcal{P}$ in a natural way as $\mathcal{P}(\hat{\mathcal{V}}(d)) = (\mathcal{P}_1(\hat{\mathcal{V}}(d)), \ldots, \mathcal{P}_N(\hat{\mathcal{V}}(d)))^T$, where for any $k \in \{1, \ldots, N\}$:

$$\mathcal{P}_k(\hat{\mathcal{V}}(d)) = \mathcal{P}(d)(q_{k-1}, q'_{k-1}).$$

Also, let $P_0 = \{\hat{\mathcal{V}}(d)|d \in \mathcal{A}\}$.

Theorem 4: Function $\mathcal{P}$ is P-contractive on P_0.

Proof: Let $d', d'' \in \mathcal{A}$, and $\hat{\mathfrak{d}}' = \hat{\mathcal{V}}(d')$, and $\hat{\mathfrak{d}}'' = \hat{\mathcal{V}}(d'')$. Let $k \in \{1, \ldots, N\}$. Then, $\mathcal{P}_k(\hat{\mathfrak{d}}) = \mathcal{P}(d)(q_{k-1}, q'_{k-1})$, $\hat{\mathfrak{d}} \in \{\hat{\mathfrak{d}}', \hat{\mathfrak{d}}''\}$. Assume that the minumum of the objective function in (19) in function $\mathcal{P}(d')(q_{k-1}, q'_{k-1})$ is reached for $\rho'_{q'} = \mu$ for $\mu \in \Phi(q)$. Further, assume that the minumum of the objective function in (19) in function $\mathcal{P}(d'')(q_{k-1}, q'_{k-1})$ is reached for $\rho'_{q'} = \nu_\sigma$ for $\sigma \in \Phi(q)$. Also, let $t(\sigma) = t$ such that $q_t = \delta(q_{k-1}, \sigma)$, and $l(\sigma) = l$ such that $q_l = f(q_t)$ (to be denoted t and l). Assume that $\mathcal{P}_k(\hat{\mathfrak{d}}') \geq \mathcal{P}_k(\hat{\mathfrak{d}}'')$. Then:

$$\begin{aligned}
&\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \\
&= \left|\sum_{\sigma\in\Psi} max(\rho_\sigma - \mu_\sigma + ed'(q_t, q_l)\mu_\sigma, ed'(q_t, q_l)\rho_\sigma) - \sum_{\sigma\in\Psi} max(\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma, ed''(q_t, q_l)\rho_\sigma)\right| \\
&\leq \left|\sum_{\sigma\in\Psi} max(\rho_\sigma - \nu_\sigma + ed'(q_t, q_l)\nu_\sigma, ed'(q_t, q_l)\rho_\sigma) - \sum_{\sigma\in\Psi} max(\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma, ed''(q_t, q_l)\rho_\sigma)\right| \\
&\text{(for } \rho'_\sigma = \mu_\sigma \text{ the minimum in } \mathcal{P}_k(\hat{\mathfrak{d}}') \text{ is reached)} \\
&\leq \left|\sum_{\sigma\in\Psi} (max(\rho_\sigma - \nu_\sigma + ed'(q_t, q_l)\nu_\sigma, ed'(q_t, q_l)\rho_\sigma) - max(\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma, ed''(q_t, q_l)\rho_\sigma))\right| \\
&\leq \sum_{\sigma\in\Psi} |max(\rho_\sigma - \nu_\sigma + ed'(q_t, q_l)\nu_\sigma, ed'(q_t, q_l)\rho_\sigma) - max(\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma, ed''(q_t, q_l)\rho_\sigma)| \qquad (25)
\end{aligned}$$

Similarly, when $\mathcal{P}_k(\hat{\mathfrak{d}}') \leq \mathcal{P}_k(\hat{\mathfrak{d}}'')$, we get (25). Again, every summand in (25) has one of the following forms:

$$|\rho_\sigma - \nu_\sigma + ed'(q_t, q_l)\nu_\sigma - (\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma)| \text{ or } |ed'(q_t, q_l)\rho_\sigma - ed''(q_t, q_l)\rho_\sigma)|, \text{ where}$$

$$|\rho_\sigma - \nu_\sigma + ed'(q_t, q_l)\nu_\sigma - (\rho_\sigma - \nu_\sigma + ed''(q_t, q_l)\nu_\sigma)| = e\nu_\sigma\,|d'(q_t, q_l) - d''(q_t, q_l)| \leq e\rho_\sigma\,|d'(q_t, q_l) - d''(q_t, q_l)|$$

and

$$|ed'(q_t, q_l)\rho_\sigma - ed''(q_t, q_l)\rho_\sigma)| = e\rho_\sigma\,|d'(q_t, q_l) - d''(q_t, q_l)|$$

Hence,

$$\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \leq \sum_{\sigma\in\Psi} e\rho_\sigma\,|d'(q_t, q_l) - d''(q_t, q_l)|$$

Further, using the same reasoning as in the proof of Lemma 1, it is straightforward to show that $\mathcal{P}$ is P-contractive. ■

Lemma 4: Let $d', d'' \in \mathcal{A}$, and $\hat{\mathfrak{d}}' = \hat{\mathcal{V}}(d')$, and $\hat{\mathfrak{d}}'' = \hat{\mathcal{V}}(d'')$. For any $k \in \{1, \ldots, N\}$, there exists $m \in \{1, \ldots, N\}$ such that:

$$\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \leq e\left|\hat{\mathfrak{d}}'_m - \hat{\mathfrak{d}}''_m\right|$$

Proof: Analogue to the proof of Lemma 2. ■

Theorem 5: For any $\hat{\mathfrak{d}}^0 \in P_0$, the sequence

$$\hat{\mathfrak{d}}^{n+1} = \mathcal{P}(\hat{\mathfrak{d}}^n), \ \ n = 0, 1, \ldots$$

converges to the only fixed point of $\mathcal{P}$ in P_0, $\hat{\mathfrak{d}}^*$, and the error of convergence is given componentwise ($k \in \{1, \ldots, N\}$) as:

$$|\hat{\mathfrak{d}}_k^n - \hat{\mathfrak{d}}_k^*| \leq (1-e)^{-1} e^n, \ \ n = 1, 2, \ldots$$

Proof: Analogue to the proof of Theorem 3. ■

The objective function in (19) is nonlinear, but fortunately transformable into a linear one by introducing additional variables y_σ.

$$\text{Minimize} \sum_{\sigma \in \Psi} y_\sigma \tag{26}$$

subject to

$$\rho_\sigma - \rho'_\sigma + c_i \rho'_\sigma \leq y_\sigma, \qquad \sigma \in \Psi$$

$$c_i \rho_\sigma \leq y_\sigma, \qquad \sigma \in \Psi$$

where, for $\sigma \in \Psi, \ c_i = e \cdot d(q_i, q'_i)$ s.t. $q_i = \delta(q, \sigma)$

$$\frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha} = \frac{\rho'_\sigma}{\sum\limits_{\alpha \in \Psi_u} \rho'_\alpha}, \qquad \sigma \in \Psi_u,$$

$$\frac{\sum_{\alpha \in \Psi_u} p_\alpha}{p_\sigma} \rho'_\sigma + \sum_{\alpha \in \Psi_c} \rho'_\alpha \leq 1, \quad \sigma \in \Psi_c,$$

$$0 \leq \rho'_\sigma \leq 1, \qquad \sigma \in \Psi,$$

$$\sum_{\alpha \in \Psi} \rho'_\alpha = 1$$

We now present the iterative algorithm for finding the fixed point of function $\mathcal{P}$.

Let $d^0(q, q') = 0$ for all $q \in Q, q' = f(q) \in Q'$. The distance $d^n(q, q')$ between the states of

$q \in Q$ and $q' = f(q) \in Q'$ in the n-th iteration ($n > 0$) is given as:

$$\text{Minimize} \sum_{\sigma \in \Psi} y_\sigma \tag{27}$$

subject to

$$\rho_\sigma - \rho'_\sigma + c_i \rho'_\sigma \leq y_\sigma, \qquad \sigma \in \Psi$$

$$c_i \rho_\sigma \leq y_\sigma, \qquad \sigma \in \Psi$$

where, for $\sigma \in \Psi,\ c_i = e \cdot d^{n-1}(q_i, q'_i)$ s.t. $q_i = \delta(q, \sigma)$

$$\frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha} = \frac{\rho'_\sigma}{\sum\limits_{\alpha \in \Psi_u} \rho'_\alpha}, \qquad \sigma \in \Psi_u,$$

$$\frac{\sum_{\alpha \in \Psi_u} p_\alpha}{p_\sigma} \rho'_\sigma + \sum_{\alpha \in \Psi_c} \rho'_\alpha \leq 1, \quad \sigma \in \Psi_c,$$

$$0 \leq \rho'_\sigma \leq 1, \qquad \sigma \in \Psi,$$

$$\sum_{\alpha \in \Psi} \rho'_\alpha = 1$$

After the n-th iteration, the value of decision variables ρ'_σ that represent the unknown transition probabilities, are such that that the distance between the (initial states of) systems G_2 and G'_2 is within $(1-e)^{-1}e^n$ of the minimal distance between the two systems (in our pseudometric). We would like to remind a reader that the aforementioned results hold for $e \in (0, 1)$.

We now summarize the presented algorithm and give a brief complexity analysis.

1) First, we modify the classical algorithm for finding the supremal controllable sublanguage. The automaton G_s, the synchronous product of the nonprobabilistic automata underlying G_p and G_r is constructed. While constructing the product, the classical controllability conditions are checked for each state. If the conditions are satisfied for each state of the product, then $G = G_s$, and go to 2. If there is at least one state of the product for which the classical conditions do not hold, the rest of the algorithm for finding the (reachable and deadlock-free) supremal controllable sublanguage is then applied. Let (reachable and deadlock-free) DES $G = (Q, \Sigma, \delta, q_0)$ represent this supremal controllable language.

2) Let G_1, G_2, and G'_2 be defined as previously in this section. Check the equalities and inequalities of the Theorem 1 for each state: if they are satisfied, supervisor exists, and G_2 is the optimal approximation. If not, then let $d^0(q, q') = 0$ for all $q \in Q, q' = f(q) \in Q'$. The distance $d^n(q, q')$ between the states of $q \in Q$ and $q' = f(q) \in Q'$ in the n-th iteration ($n > 0$) is given by (27). For each of the state of G_2 (typically, the number of states of G_2 is much smaller than

$|Q_p| \cdot |Q_r|$), simplex method can be used to efficiently solve the linear programming problem (27). The worst-case time complexity of simplex method is exponential in the number of decision variables. In our case, the number of decision variables is twice the number of events possible from the state q. As this number is typically small in practical applications, this exponential complexity does not generally present a limitation of the algorithm. Furthermore, the number of iterations needed to reach the accuracy of ϵ is $\lceil (\log_e \epsilon + \log_e (1-e)) \rceil$.

C. Example

For a plant G_p as depicted in Fig. 1, there does not exist a probabilistic supervisor V_p such that $G(V_p/G_p) = G_r$. PDESs G_1 and G_2 in Figure 2 represent the supremal controllable sublanguage $L(G_p) \cap L(G_r)$ (with respect to G_p) with probability distributions p and p_2 given as in (17) and (18), respectively. For PDES G_2, let ρ_q be the probability distribution induced by the state $q \in Q$ and, for PDES G_2', let $\rho'_{q'}$ be the probability distribution induced by the state $q' \in Q'$. Also, we will write $\rho_{q,\sigma}$ instead of $\rho_q(\sigma, q_i)$, and $\rho'_{q',\sigma}$ instead of $\rho'_{q'}(\sigma, q_j)$.

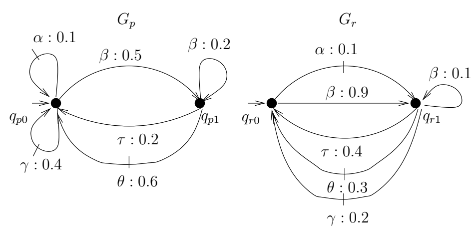

Fig. 1. Plant G_p, and requirements specification G_r

At n-th iteration, distances $d^n(q_0, q'_0), d^n(q_1, q'_1)$, and $d^n(q_2, q'_2)$ are calculated as follows:

$$d^n(q_0, q'_0) = \text{Minimize } (y_{q_0,\alpha} + y_{q_0,\beta})$$

subject to

$$\rho_{q_0,\alpha} - \rho'_{q'_0,\alpha} + e \cdot d^{n-1}(q_1, q'_1)\rho'_{q'_0,\alpha} \leq y_{q_0,\alpha}, \quad e \cdot d^{n-1}(q_1, q'_1)\rho_{q_0,\alpha} \leq y_{q_0,\alpha},$$

$$\rho_{q_0,\beta} - \rho'_{q'_0,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_0,\beta} \leq y_{q_0,\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_0,\beta} \leq y_{q_0,\beta},$$

$$\frac{p(q_0, \beta)}{p(q_0, \alpha)}\rho'_{q_0,\alpha} + \rho'_{q_0,\alpha} \leq 1,$$

$$0 \leq \rho'_{q'_0,\alpha} \leq 1, \quad 0 \leq \rho'_{q'_0,\beta} \leq 1, \quad \rho'_{q'_0,\alpha} + \rho'_{q'_0,\beta} = 1.$$

$$d^n(q_1, q'_1) = \text{Minimize } (y_{q_1,\beta} + y_{q_1,\gamma})$$

subject to

$$\rho_{q_1,\beta} - \rho'_{q'_1,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_1,\beta} \leq y_{\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_1,\beta} \leq y_{q_1,\beta},$$

$$\rho_{q_1,\gamma} - \rho'_{q'_1,\gamma} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_1,\gamma} \leq y_{\gamma}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_1,\gamma} \leq y_{q_1,\gamma},$$

$$\frac{p(q_1, \beta)}{p(q_1, \gamma)}\rho'_{q_0,\gamma} + \rho'_{q_1,\gamma} \leq 1,$$

$$0 \leq \rho'_{q'_1,\beta} \leq 1, \quad 0 \leq \rho'_{q'_1,\gamma} \leq 1, \quad \rho'_{q'_1,\beta} + \rho'_{q'_1,\gamma} = 1.$$

$$d^n(q_2, q'_2) = \text{Minimize } (y_{q_2,\beta} + y_{q_2,\theta} + y_{q_2,\tau})$$

subject to

$$\rho_{q_2,\beta} - \rho'_{q'_2,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_2,\beta} \leq y_{\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_2,\beta} \leq y_{q_2,\beta},$$

$$\rho_{q_2,\theta} - \rho'_{q'_2,\theta} + e \cdot d^{n-1}(q_0, q'_0)\rho'_{q'_2,\theta} \leq y_{\gamma}, \quad e \cdot d^{n-1}(q_0, q'_0)\rho_{q_2,\theta} \leq y_{q_2,\theta},$$

$$\frac{p(q_2, \tau)}{p(q_2, \tau) + p(q_2, \beta)} = \frac{\rho'_{q'_2,\tau}}{\rho'_{q'_2,\tau} + \rho'_{q'_2,\beta}}, \quad \frac{p(q_2, \beta) + p(q_2, \tau)}{p(q_1, \theta)}\rho'_{q_0,\theta} + \rho'_{q_1,\theta} \leq 1,$$

$$0 \leq \rho'_{q'_2,\beta} \leq 1, \quad 0 \leq \rho'_{q'_2,\theta} \leq 1, \quad 0 \leq \rho'_{q'_2,\tau} \leq 1, \quad \rho'_{q'_2,\beta} + \rho'_{q'_2,\theta} + \rho'_{q'_2,\tau} = 1.$$

After 20 iterations, we find that the closest behaviour (for $e = 0.5$) achievable with probabilistic control is as given in Fig. 2.

V. CONCLUSIONS

Our goal is to solve a classical problem of nearest approximation in the framework of probabilistic control of PDES. We suggest two algorithms for the calculation of the distances between the states of a probabilistic generator used to model PDES in the chosen pseudometric.

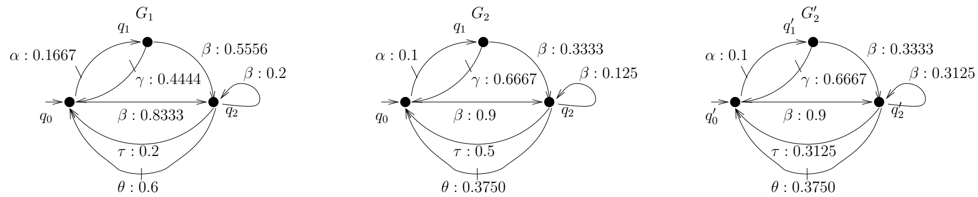

Fig. 2. PDESs G_1, G_2 and the closest approximation G_2'

Then, we suggest the straightforward modification of the iterative algorithm to be used to minimize the distance (in this pseudometric) between the required behaviour of a system and its achievable behaviour.

Although we know the rate of convergence of the algorithm to the minimal distance, we would like to investigate how unknown, desired probabilities change as the distance converges. Also, the question of uniqueness of the closest approximation remains open as well as probabilistic control with marking. Further, it would be interesting to compare the distance between the required behavior and its nearest approximation achieved using probabilistic control with the distance between the required behavior and its nearest approximation obtained using deterministic control only.

References

[1] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*. Homewood, IL, USA: Richard D. Irwin, Inc., and Aksen Associates, Inc., 1993.

[2] M. O. Rabin, "Probabilistic automata," *Information and Control*, vol. 6, no. 3, pp. 230–245, September 1963.

[3] M. Molloy, "Performance analysis using stochastic Petri Nets," *IEEE Transactions on Computers*, vol. C-39, no. 9, pp. 913–917, 1982.

[4] V. Garg, "An algebraic approach to modeling probabilistic discrete event systems," in *Proceedings of 31st IEEE Conference on Decision and Control*, Tucson, AZ, USA, Dec. 1992, pp. 2348–2353.

[5] ——, "Probabilistic languages for modeling of DEDS," in *Proceedings of 26th Conference on Information Sciences and Systems*, vol. 1, Princeton, NJ, Mar. 1992, pp. 198–203.

[6] H. Mortazavian, "Controlled stochastic languages," in *Proceedings of 31st Annual Allerton Conference on Communications, Control, and Computing*, Urbana, Illinois, 1993, pp. 938–947.

[7] V. S. Borkar, *Topics in controlled Markov chains*. New York: Wiley, 1991.

[8] R. Kumar and V. Garg, "Control of stochastic discrete event systems: Existence," in *Proceedings of 1998 International Workshop on Discrete Event Systems*, Cagliari, Italy, august 1998, pp. 24–29.

[9] M. Lawford and W. Wonham, "Supervisory control of probabilistic discrete event systems," in *Proceedings of the 36th IEEE Midwest Symposium on Circuits and Systems*, vol. 1. IEEE, Aug. 1993, pp. 327–331.

[10] R. Kumar and V. Garg, "Control of stochastic discrete event systems modeled by probabilistic languages," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, pp. 593–606, Apr. 2001.

[11] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite state systems," in *Proceedings of 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, Dec. 2007, pp. 5930–5935.

[12] Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *International Journal of Control*, vol. 80, no. 8, pp. 1271–1290, 2007.

[13] S. Postma and M. Lawford, "Computation of probabilistic supervisory controllers for model matching," in *Proceedings of Allerton Conference on Communications, Control, and Computing*, 2004.

[14] V. Pantelic, S. Postma, and M. Lawford, "Supervisory control of probabilistic discrete event systems," *IEEE Transactions on Automatic Control*, 2009, accepted subject to revision.

[15] C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski, "Controller synthesis for probabilistic systems," in *Proceedings of the IFIP International Conference on Theoretical Computer Science*, J.-J. Lévy, E. W. Mayr, and J. C. Mitchell, Eds. Kluwer, 2004, pp. 493–506.

[16] A. Kučera and O. Stražovský, "On the controller synthesis for finite-state Markov decision processes," *Fundamenta Informaticae*, vol. 82, no. 1-2, pp. 141–153, 2008.

[17] Y. Deng, T. Chothia, C. Palamidessi, and J. Pang, "Metrics for action-labelled quantitative transition systems," *Electronic Notes in Theoretical Computer Science*, vol. 153, no. 2, pp. 79–96, 2006, also appeared in *Proceedings of the 3rd Workshop on Quantitative Aspects of Programming Languages*.

[18] V. Pantelic and M. Lawford, "Towards optimal supervisory control of probabilistic discrete event systems," in *Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems*, Bari, Italy, June 2009, accepted, available at http://www.cas.mcmaster.ca/ pantelv/ifac09optimaldraft.pdf.

[19] G. Barrett and S. Lafortune, "Using bisimulation to solve discrete event control problems," in *Proceedings of the 1997 American Control Conference*, Albuquerque, NM, June 1997, pp. 2337–2341.

[20] K. G. Larsen and A. Skou, "Bisimulation through probabilistic testing," *Information and Computation*, vol. 94, no. 1, pp. 1–28, 1991.

[21] A. Sokolova and E. d. Vink, "Probabilistic automata: system types, parallel composition and comparison," in *Validation of Stochastic Systems: A Guide to Current Research*, C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle, Eds. LNCS 2925, 2004, pp. 1–43.

[22] A. Giacalone, C. Jou, and S. Smolka, "Algebraic reasoning for probabilistic concurrent systems," in *M. Broy and C.B. Jones, eds, Proceedings of the Working Conference on Programming Concepts and Methods*. Sea of Gallilee, Israel: North-Holland, 1990, pp. 443–458.

[23] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labeled Markov systems," in *CONCUR*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and S. Mauw, Eds., vol. 1664. Springer, 1999, pp. 258–273.

[24] A. Arnold, *Finite Transition Systems*. Prentice Hall, 1994.

[25] L. Kantorovich, "A new method for solving some classes of extremal problems," *Comptes Rendus (Doklady) Acad. Sci. USSR*, vol. 28, pp. 211–214, 1940.

[26] J. E. Hutchinson, "Fractals and self-similarity," *Indiana University Mathematics Journal*, vol. 30, pp. 713–747, 1981.

[27] D. Kozen, "A probabilistic PDL," *Journal of Computer and System Sciences*, vol. 30, no. 2, pp. 162–178, 1985.

[28] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled Markov processes," *Theor. Comput. Sci.*, vol. 318, no. 3, pp. 323–354, 2004.

[29] ——, "Approximating labeled Markov processes," in *Logic in Computer Science*, 2000, pp. 95–106. [Online]. Available: citeseer.ist.psu.edu/article/desharnais00approximating.html

[30] J. Desharnais, R. Jagadeesan, V. Gupta, and P. Panangaden, "The metric analogue of weak bisimulation for probabilistic processes," in *LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 413–422.

[31] F. van Breugel and J. Worrell, "An algorithm for quantitative verification of probabilistic transition systems," in *CONCUR*, ser. Lecture Notes in Computer Science, K. G. Larsen and M. Nielsen, Eds., vol. 2154. Springer, 2001, pp. 336–350.

[32] ——, "A behavioural pseudometric for probabilistic transition systems," *Theor. Comput. Sci.*, vol. 331, no. 1, pp. 115–142, 2005.

[33] E. P. de Vink and J. J. M. M. Rutten, "Bisimulation for probabilistic transition systems: a coalgebraic approach," *Theor. Comput. Sci.*, vol. 221, no. 1-2, pp. 271–293, 1999.

[34] C. Baier and M. Kwiatkowska, "Domain equations for probabilistic processes," in *Proc. EXPRESS'97, Santa Margherita Ligure*, ser. Electronic Notes in Theoretical Computer Science, vol. 7. Elsevier, 1997.

[35] F. van Breugel and J. Worrell, "Towards quantitative verification of probabilistic transition systems," in *ICALP*, ser. Lecture Notes in Computer Science, F. Orejas, P. G. Spirakis, and J. van Leeuwen, Eds., vol. 2076. Springer, 2001, pp. 421–432.

[36] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. New York, New York, USA: Academic Press, Inc., 1970.