Formal Controller Synthesis for Markov Jump Linear Systems with Uncertain Dynamics *

Luke Rickard[1 [0000−0002−9192−9186]], Thom Badings[2 [0000−0002−5235−1967]], Licio Romao[1 [0000−0002−5716−6162]], and Alessandro Abate[1 [0000−0002−5627−9093]]

[1] University of Oxford, Oxford, UK
[2] Radboud University, Nijmegen, the Netherlands

Abstract. Automated synthesis of provably correct controllers for cyber-physical systems is crucial for deployment in safety-critical scenarios. However, hybrid features and stochastic or unknown behaviours make this problem challenging. We propose a method for synthesising controllers for Markov jump linear systems (MJLSs), a class of discrete-time models for cyber-physical systems, so that they certifiably satisfy probabilistic computation tree logic (PCTL) formulae. An MJLS consists of a finite set of stochastic linear dynamics and discrete jumps between these dynamics that are governed by a Markov decision process (MDP). We consider the cases where the transition probabilities of this MDP are either known up to an interval or completely unknown. Our approach is based on a finite-state abstraction that captures both the discrete (mode-jumping) and continuous (stochastic linear) behaviour of the MJLS. We formalise this abstraction as an interval MDP (iMDP) for which we compute intervals of transition probabilities using sampling techniques from the so-called 'scenario approach', resulting in a probabilistically sound approximation. We apply our method to multiple realistic benchmark problems, in particular, a temperature control and an aerial vehicle delivery problem.

Keywords: Markov Jump Linear Systems · Stochastic Models · Uncertain Models · Robust Control Synthesis · Temporal logic · Safety Guarantees

1 Introduction

In a world where autonomous cyber-physical systems are increasingly deployed in safety-critical settings, it is important to develop methods for certifiable control of these systems [32]. Cyber-physical systems are characterised by the coupling of digital (discrete) computation with physical (continuous) dynamical components. This results in a *hybrid system*, endowed with different discrete modes of operation, each of which is characterised by its own continuous dynamics [35]. Ensuring that these hybrid systems meet complex and rich formal specifications when controlled is an important yet challenging goal.

* This work was supported by funding from the EPSRC AIMS CDT EP/S024050/1, and by NWO grant NWA.1160.18.238 (PrimaVera).

Formal controller synthesis Often, these specifications cannot be expressed as classical control-theoretic objectives, which by and large relate to stability and convergence, or invariance and robustness [8]. Instead, these requirements can be expressed in a temporal logic, which is a rich language for specifying the desired behaviour of dynamical systems [44]. In particular, probabilistic computation tree logic (PCTL, [27]) is widely used to define temporal requirements on the behaviour of probabilistic systems. For example, in a building temperature control problem, a PCTL formula can specify that, with at least 75% probability, the temperature must stay within the range $22-23°\mathrm{C}$ for 10 minutes. Leveraging probabilistic verification tools [7], it is of interest to synthesise a controller that ensures the satisfaction of such a PCTL formula for the model under study [26].

Markov jump linear systems Markov jump linear systems (MJLSs) [20] are a well-known class of stochastic, hybrid models suitable for capturing the behaviour of cyber-physical systems [35]. An MJLS consists of a finite set of linear dynamics (also called *operational modes*), where jumps between these modes are governed by a Markov chain (MC) or, if jumping between the modes can be controlled, by a Markov Decision Process (MDP). Despite each mode having linear (though possibly stochastic) dynamics, the overall dynamics are non-linear due to the jumping between modes. MJLSs have been used to model, among other things, networked control systems, where the different operation modes relate to specific packet losses or to distinct discrete configurations [29,42].

Uncertainty in MJLSs We consider a rich class of discrete-time MJLSs with two sources of uncertainty. First, the continuous dynamics in each mode are affected by an additive stochastic process noise, e.g., due to inaccurate modelling or wind gusts affecting a drone [10]. We only assume sampling-access to the noise, rather than full knowledge of its probability distribution, allowing us to provide probably approximately correct (PAC) guarantees on the behaviour of the MJLS. Second, similar to [42], we assume that the transition probabilities of the Markov jump process are not precisely known. However, unlike [42], we consider two different semantics for this uncertainty: either (1) transition probabilities between modes are given by intervals; or (2) these probabilities are not known at all [31,36]. More details on the considered model are in Sect. 2.

Problem statement Several MJLS control problems have been studied, such as stability [12,56], H_∞-controller design [19,21,22,55], and optimal control [30,54]. However, limited research has been done for more complex tasks expressed in, for example, PCTL. In this paper, we thus solve the following problem. Given an MJLS subject to uncertainty in both its continuous dynamics (via additive noise of an unknown distribution) and its discrete behaviour (uncertain Markov jumps), compute a provably correct controller that satisfies a given PCTL formula.

Abstractions of MJLSs We develop a new technique for abstracting MJLSs by extending methods introduced for linear non-hybrid systems in [4]. In line with [4], we capture the stochastic noise affecting the continuous dynamics by means of

transition probability *intervals* between the discrete states of the abstraction. We compute these intervals using sampling techniques from the *scenario approach* [16] and leverage the tighter theoretical bounds developed in [49]. We thus formalise the resulting abstract model as an interval MDP (iMDP), which is an MDP with transition probabilities given as intervals [24]. Different from [4], we also newly capture the discrete mode jumps in the abstract iMDP.

Controller synthesis We use the state-of-the-art verification tool PRISM [33] to synthesise a policy on the abstract iMDP that satisfies a given PCTL specification. Leveraging results from the scenario approach, we refine this policy into a controller for the MJLS with PAC guarantees on the satisfaction of the specification.

Contributions Our main contribution is a framework to synthesise provably-correct controllers for discrete-time MJLSs given general PCTL specifications, based on iMDP abstractions of the MJLSs. Previous work in this area has been limited to linear time-invariant dynamics, and to simpler reach-avoid specifications [4]. We thus extend earlier techniques by developing new methods for a broader class of hybrid models (MJLSs) and for general PCTL formulae. In line with previous work, we propose a semi-algorithm based on iterative refinements of our model, meaning that a synthesised controller will satisfy the required formula, but the inability to find such a controller does not imply the non-existence of one. Technically, we newly show how to capture both the continuous and discrete dynamics of the MJLS in the abstract iMDP model. In particular, our methods are applicable to MJLSs where the stochastic noise in the continuous dynamics and that in the transition probabilities of the Markov jump process are unknown.

Related Work

Techniques for providing safety guarantees for dynamical systems can largely be split into two approaches, respectively called *abstraction-free* and *abstraction-based* [35].

Abstraction-free methods derive safety guarantees without the need to create simpler abstract models. For example, barrier functions [37,43,48] can be used to certify the existence of control inputs that keep the system within safe states. Another approach is that of (probabilistic) reachability computation [3,41], where the goal is to evaluate if the system will reach a certain state over a given horizon.

Abstraction-based methods [8,51] analyse a simpler model of the system, formally shown to be related to the concrete model, and thus allow to transfer the obtained results (safety guarantees, or synthesised policies) back to the original model. Various approaches exist for creating abstractions of different forms, including the celebrated counterexample-guided abstraction/refinement approach [18] and, relevant for this work, a few involve abstractions as Markov models [1,3,5,6,17,50].

Related to the approaches detailed above is robust control, where the goal is to compute a controller that achieves some task while being robust against disturbances. Robust control techniques for MJLSs have been studied in [9,13,52].

2 Foundations and Problem Statement

2.1 Markov Decision Processes

A Markov decision process (MDP) is a tuple $\mathcal{M}=(\mathcal{S},\mathcal{A},s_I,P)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $s_I \in \mathcal{S}$ is the initial state, and $P\colon \mathcal{S}\times\mathcal{A} \rightharpoonup Dist(\mathcal{S})$ is a (partial) probabilistic transition function, with $Dist(\mathcal{S})$ the set of all probability distributions over $\mathcal{S}$ [7]. We call a tuple (s,a,s') with probability $P(s,a)(s')>0$ a *transition*. We write $\mathcal{A}(s)\subseteq\mathcal{A}$ for the actions enabled in state s. A Markov chain (MC) is an MDP such that $|\mathcal{A}(s)|=1,\forall s\in S$. We consider time-dependent deterministic (or pure) policies, $\pi\colon \mathcal{S}\times\mathbb{N}\to\mathcal{A}$, which map states $s\in\mathcal{S}$ and time steps $k\in\mathbb{N}$, to actions $a\in\mathcal{A}(s)$. The set of all policies for MDP $\mathcal{M}$ is denoted by $\Pi_{\mathcal{M}}$.

Interval Markov decision processes (iMDPs) extend regular MDPs with uncertain transition probabilities [24]. An iMDP is a tuple $\mathcal{M}_{\mathbb{I}}=(\mathcal{S},\mathcal{A},s_I,\mathcal{P})$, where the states and actions are defined as for MDPs, and $\mathcal{P}\colon \mathcal{S}\times\mathcal{A}\rightharpoonup 2^{Dist(\mathcal{S})}$ maps states and actions to a set of distributions over successor states. Specifically, each $\mathcal{P}(s,a)(s')$ is an *interval* of the form $[\underline{p},\overline{p}]$, with $\underline{p},\overline{p}\in(0,1],\underline{p}\leq\overline{p}$. Intuitively, an iMDP encompasses a set of MDPs differing only in their transition probabilities: fixing an allowable probability distribution in the set $\mathcal{P}(s,a)$ for every state-action pair (s,a) (denoted $P\in\mathcal{P}$ for brevity) results in an MDP, denoted by $\mathcal{M}_{\mathbb{I}}^{P}$.

2.2 Markov Jump Linear Systems

Let $\mathcal{Z}=\{z_1,...,z_N\}$ be a finite set of discrete modes. Consider the collections of matrices $A=(A_1,...,A_N)$, $A_i\in\mathbb{R}^{n\times n}$, and $B=(B_1,...,B_N)$, $B_i\in\mathbb{R}^{n\times m}$; and of vectors $q=(q_1,...,q_N),q_i\in\mathbb{R}^n$. A discrete-time MJLS model $\mathfrak{J}$ comprises continuous and discrete dynamics. Each triple (A_i,B_i,q_i) defines a linear dynamical system, with discrete-time dynamics in (1a). The discrete jumps between the N modes in $\mathcal{Z}$ are governed by an MDP $(\mathcal{Z},\mathcal{B},z_I,T)$ with *switching actions* $\mathcal{B}=\{1,...,M\}$, and *mode switch* transition function $T\colon \mathcal{Z}\times\mathcal{B}\rightharpoonup Dist(\mathcal{Z})$. At any time $k\in\mathbb{N}$, we denote the continuous state by $x(k)\in\mathcal{X}\subseteq\mathbb{R}^n$ ($\mathcal{X}$ bounded), and the discrete mode by $z(k)\in\mathcal{Z}$. Given initial state $x(0)\in\mathcal{X},z(0)\in\mathcal{Z}$, the (hybrid) state (x,z) is computed as

$$\mathfrak{J}\colon \begin{cases} x(k+1)=A_{z(k)}x(k)+B_{z(k)}u(k)+q_{z(k)}+w_{z(k)}(k) & \text{(1a)}\\ z(k+1)\sim T(z(k),b(k)), & \text{(1b)}\end{cases}$$

where $u(k)\in\mathcal{U}\subseteq\mathbb{R}^m$ is the control input to the continuous dynamics, and $b(k)\in\mathcal{B}$ is the discrete (MDP) switching action. Note, for each mode $z\in\mathcal{Z}$, the corresponding continuous dynamics are affected by an additive stochastic process noise w_z, with a (potentially) unknown distribution. The distribution of the noise w_z is not required to be the same across different modes, but $\{w_z(k)\}_{k\in\mathbb{N}}$ must be an i.i.d. stochastic process having density with respect to the Lebesgue measure, and independent across modes. Importantly for our setting, the input u and switch b are *jointly determined* by a feedback controller (namely, a policy for the MJLS) of the following form.

Definition 1. *A time-dependent feedback* controller $F\colon \mathcal{X}\times\mathcal{Z}\times\mathbb{N}\to\mathcal{U}\times\mathcal{B}$ *is a function that maps a continuous state $x\in\mathcal{X}$, discrete mode $z\in\mathcal{Z}$, and time step $k\in\mathbb{N}$, to a continuous control input $u\in\mathcal{U}$ and discrete switch $b\in\mathcal{B}$.*

Example 1. Consider a temperature regulation problem inspired by [3], in which a portable fan heater and a portable radiator are used to heat a two-room building. We define two modes $\mathcal{Z}=\{1,2\}$, relating to the fan heater being in room 1 or 2 respectively (and the radiator in the other room). Swapping the heat sources between rooms is modelled by an MDP with actions $\mathcal{B}=\{0,1\}$, relating to leaving or switching the heaters. Each of these mode-switching actions fails with some probability. This problem is naturally modelled as an MJLS with the matrices

$$\begin{aligned} A_{\{1,2\}} &= \begin{bmatrix} 1-b_1-a_{12} & a_{12} \\ a_{21} & 1-b_2-a_{21} \end{bmatrix}, \\ B_1 &= \begin{bmatrix} k_f & 0 \\ 0 & k_r \end{bmatrix},\ B_2 = \begin{bmatrix} k_r & 0 \\ 0 & k_f \end{bmatrix},\ q_{\{1,2\}} = \begin{bmatrix} b_1 x_a \\ b_2 x_a \end{bmatrix}, \end{aligned} \tag{2}$$

where the state $x=[T_1,T_2]^\top \in \mathbb{R}^2$ models the room temperatures, and the power of both heaters can be adjusted within the range $u\in[0,1]^2$ (the extrema denoting being fully on and off). In Sect. 6, we perform a numerical experiment with this MJLS. □

2.3 Probabilistic Computation Tree Logic

Probabilistic computation tree logic (PCTL) depends on the following syntax [7]:

$$\begin{aligned} &\Phi ::= true \mid p \mid \neg\Phi \mid \Phi\wedge\Phi \mid \mathsf{P}_{\sim\lambda}(\psi) \\ &\psi ::= \Phi \mathsf{U} \Phi \mid \Phi \mathsf{U}^{\leq K} \Phi \mid \mathsf{X}\Phi. \end{aligned} \tag{3}$$

Here, $\sim\in\{<,\leq,\geq,>\}$ is a comparison operator and $\lambda\in[0,1]$ a probability threshold; PCTL formulae Φ are state formulae, which can in particular depend on path formulae ψ. Informally, the syntax consists of state labels $p\in AP$ in a set of atomic propositions AP, propositional operators negation $\neg$ and conjunction $\wedge$, and temporal operators until U, bounded until $\mathsf{U}^{\leq K}$, and next X. The probabilistic operator $\mathsf{P}_{\sim\lambda}(\psi)$ requires that paths generated from the initial conditions satisfy a path formula ψ with total probability exceeding (or below, depending on $\sim$) some given threshold λ.

An MJLS $\mathfrak{J}$ with a controller F induces a stochastic process on the hybrid state space $\mathcal{X}\times\mathcal{Z}$. Let $L_{\mathcal{J}}\colon \mathcal{X}\times\mathcal{Z}\to 2^{AP}$ be a labelling from hybrid states to a subset of labels. Recall that the noise affecting the continuous dynamics in (1) has density with respect to the Lebesgue measure. We assume for each label that the set $\{x\in\mathcal{X}\colon p\in L_{\mathcal{J}}(x,z), z\in\mathbb{Z}\}\subseteq\mathcal{X}$ of continuous states with label p is measurable. We follow the same semantics as used in [34] for stochastic hybrid systems, i.e., the (initial) state $x(0),z(0)$ of an MJLS $\mathfrak{J}$ satisfies a property $\Phi=\mathsf{P}_{\sim\lambda}(\psi)$ if the probability of all paths from $x(0),z(0)$ satisfies $\sim\lambda$. For brevity, we shall write this satisfaction relation as $\mathfrak{J}\models_F\Phi$. All the sets of paths $(x(0),z(0)),(x(1),z(1)),\ldots$ expressed by PCTL under the above assumptions are measurable, see, e.g., [35,46,53] for details.

For an iMDP $\mathcal{M}_{\mathbb{I}}$, the satisfaction relation $\mathcal{M}_{\mathbb{I}}\models_\pi\Phi$ defines whether a PCTL formula Φ holds true, when following policy π from the initial state(s). Formal definitions for semantics and model checking are provided in [7,27]. Recall from (4) that for iMDPs $\mathcal{M}_{\mathbb{I}}$, the threshold $\sim\lambda$ must hold under the *worst-case realization* of the probabilities $P\in\mathcal{P}$ in their intervals. That is, we are interested in synthesizing

an optimal policy $\pi^\star \in \Pi_{\mathcal{M}_\mathbb{I}}$ that maximises the probability of satisfying a path specification ψ for the *worst-case* assignment $P \in \mathcal{P}$ (which is determined by a so-called *adversary*). In other words, we seek to solve the max-min decision problem given by

$$\pi^\star = \underset{\pi \in \Pi_{\mathcal{M}_\mathbb{I}}}{\operatorname{argmax}} \min_{P \in \mathcal{P}} \lambda \quad \text{s.t. } \mathsf{P}_{\geq \lambda}(\mathcal{M}_\mathbb{I}^P \models_\pi \psi). \tag{4}$$

It is shown by [38,45], and in a much more general setting by [25], that deterministic policies suffice to obtain optimal values for iMDPs.

2.4 Problem Statement

We consider tasks encoded as a PCTL formula Φ. Our goal is to find a feedback controller F that satisfies Φ. As such, we solve the following problem.

Problem 1. Given an MJLS $\mathfrak{J}$ as in (1) and a PCTL formula Φ, find a control policy $F\colon \mathcal{X} \times \mathcal{Z} \times \mathbb{N} \to \mathcal{U} \times \mathcal{B}$, such that $\mathfrak{J} \models_F \Phi$.

In this paper, we address Problem 1 through the lens of abstractions [51], under two distinct assumptions on the mode-transition function of the MJLS.

Assumption A (Uncertain Markov jumps) *Each transition probability of the MDP $(\mathcal{Z}, \mathcal{B}, z_I, T)$ driving the jumps across modes in $\mathcal{Z}$ is known up to a certain interval $\mathcal{T} \ni T$, i.e., the Markov jump process is an iMDP $(\mathcal{Z}, \mathcal{B}, z_I, \mathcal{T})$.*

Assumption B (Unknown Markov jumps) *The Markov jumps are driven by a Markov chain (MC) for which we can measure the current mode, but the transition function (and hence its underlying graph structure) is unknown.*

For Assumption A, we will exploit the transition function of the jump process to reason over the joint probability distribution over the state $x(k)$ and mode $z(k)$. By contrast, under Assumption B, we do not know the distribution over successor modes $z(k+1)$, so reasoning over the joint distribution is not possible. Instead, our goal is to attain *robustness* against any mode changes that may occur. An overview of our abstraction-based approach to solve Problem 1 is presented in Fig. 1. We note that it may occur that the PCTL formula is not satisfiable on the abstract model. To alleviate this issue, we propose an iterative refinement of the abstraction (shown by the dashed line in Fig. 1), which we explain in more detail in Sect. 5.

3 Abstractions of Non-Hybrid Dynamical Systems

Our abstraction procedure expands on the techniques from [4,6] to make them applicable to hybrid (and probabilistic) models. We start by summarising the main contributions of these papers, while referring to [4,6] for proofs and more details.

Consider a discrete-time linear system $\mathfrak{L}$ with additive stochastic noise:

$$\mathfrak{L}\colon x(k+1) = Ax(k) + Bu(k) + q + w(k), \tag{5}$$

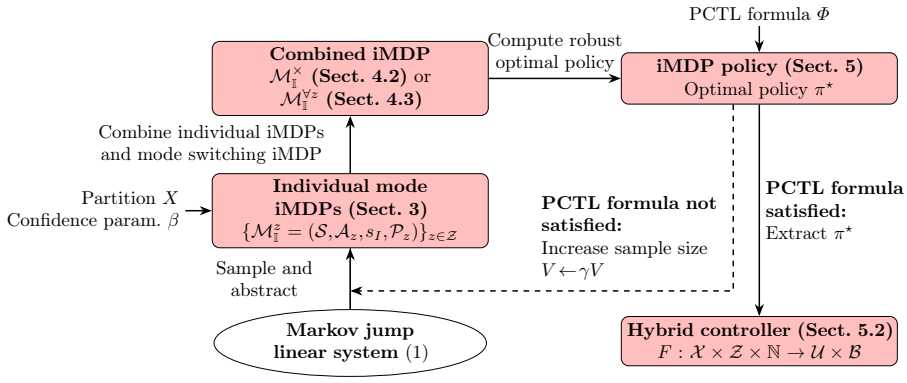

Fig. 1. Approach for synthesising a provably-correct controller for an MJLS.

where $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, q \in \mathbb{R}^n$, and $w(k)$ defines an i.i.d. stochastic process, and $x(k) \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u(k) \in \mathcal{U} \subseteq \mathbb{R}^m$ are the states and control inputs, respectively. The distribution of the noise $w(k)$ is assumed to be unknown, but instead we have access to a set $\{\delta_1, ..., \delta_V\}$ of V i.i.d. samples of $w(k)$. Note that the system in (5) reduces to an MJLS with a single mode. Given such a set of i.i.d. samples, the authors in [6] show how to construct an iMDP which, with a specified confidence level, abstracts the system in (5):

Definition 2 (β-iMDP abstraction). *Choose $\beta \in (0,1)$ and let $\{\delta_1, ..., \delta_V\}$ be a collection of samples from the noise distribution affecting the dynamics in (5). An iMDP $\mathcal{M}_{\mathbb{I}} = (\mathcal{S}, \mathcal{A}, s_I, \mathcal{P})$ is a β-iMDP abstraction if for every PCTL formula Φ and for every policy $\pi \in \Pi_{\mathcal{M}_{\mathbb{I}}}$, there exists a feedback control $F \colon \mathcal{X} \times \mathbb{N} \to \mathcal{U}$ such that, for any initial condition $x(0)$, we have that*

$$\mathbb{P}^V \left\{ (\mathcal{M}_{\mathbb{I}} \models_{\pi} \Phi) \implies (\mathfrak{L} \models_{F} \Phi) \right\} \geq 1 - \beta, \tag{6}$$

where s_I is the initial state of the β-iMDP associated with continuous state $x(0)$, and $\mathbb{P}^V$ is the product probability measure induced by the sample set $\{\delta_1, ..., \delta_V\}$.

We remark that $\mathbb{P}^V$ is the product probability measure corresponding with sampling a set $\{\delta_1, ..., \delta_V\}$ of $V \in \mathbb{N}$ samples of the noise $w(k)$ in (5) (see, e.g., [14] for details). Def. 2 states that, with a confidence of at least $1 - \beta$, the satisfaction of a formula on the abstract iMDP implies the existence of a feedback controller that allows the satisfaction of the same formula on the concrete model. The confidence bound accounts for the inherent statistical error caused by constructing the iMDP based on a finite set of noise samples only. The iMDP abstraction allows us to synthesise correct-by-design feedback controllers for continuous-state dynamical systems [40], by utilising policies designed for a discrete-state model. Note that Def. 2 applies to general PCTL formulas, while [6] only considers reach-avoid properties (a subset of PCTL).

Definition 3 (Partition). *A partition $X = \{X_1, \ldots, X_p\}$ is an ordered set of subsets of $\mathcal{X}$ such that $\mathcal{X} = \bigcup_{i=1}^{p} X_i$, and $X_i \bigcap X_j = \emptyset, \forall i, j \in \{1, ..., p\}, i \neq j$.*

Papers [4,6] show how to generate β-iMDP abstractions by combining partitioning of the state space, backward reachability computation, and the scenario approach theory [14]. To this end, these papers create an iMDP abstraction $(\mathcal{S},\mathcal{A},s_I,\mathcal{P})$ of the continuous-state dynamics using the following procedure:

- The set of states $\mathcal{S}=\{s_1,...,s_p\}\cup\{s^\star\}$ consists of elements associated with a partition X of the state space. This correspondence is given by the quotient mapping induced by the equivalence relation of the partition (see, e.g., [51]).
- The action space $\mathcal{A}=\{a_1,...,a_q\}$, where each action $a\in\mathcal{A}$ is associated with a target point $d\in\mathcal{X}$ in the continuous state space (a convenient choice is to define each target d as the centre of an element $X_i\in X$ of the partition).
- To decide which actions are enabled at a given state of the abstraction, backward reachable set computations are employed. More specifically, we let

$$\mathcal{R}^{-1}(a)=\{x\in\mathbb{R}^n\,|\,d=Ax+Bu+q,u\in\mathcal{U}\} \tag{7}$$

  be the backward reachable set of the target point d associated with the action $a\in\mathcal{A}$. Action a is enabled in state $s\in\mathcal{S}$ if and only if its corresponding element $X_i\in X$ is contained in $\mathcal{R}^{-1}(a)$. Mathematically, we have that

$$\mathcal{A}(s)=\big\{a\in\mathcal{A}\,|\,X_i\subseteq\mathcal{R}^{-1}(a)\big\}. \tag{8}$$

- The initial state s_I of the iMDP is defined by the element of the partition to which the initial state of the continuous dynamics belongs.
- The probability intervals $\mathcal{P}(s,a_j)(s_i)$ of the abstract iMDP can be efficiently computed using the scenario approach [15,49], or using statistical inequalities such as Hoeffding's bound [11].

To show that this procedure indeed yields a β-iMDP abstraction as per Def. 2, we also invoke the following key result from [6]:

Theorem 1 (iMDP abstraction of stochastic linear systems [6]). *Let X be a partition of the state space, then for any $\beta\in(0,1)$ and sample set $\{\delta_1,...,\delta_V\}$, the procedure above yields a β-iMDP abstraction for the dynamics in* (5).

We provide an intuitive proof outline here, while referring to [6] for the full proof. Consider state s, with an associated continuous state $x(k)$; successor state s', with associated partition X_i; action a, with an associated feedback controller F. The true probability of transitioning from s to s', under action a is defined as

$$\mathrm{P}^\star(s,a)(s')=\int_{\mathbb{R}^n}\mathbb{1}_{X_i}(Ax(k)+BF(x(k),k)+q+\xi)\mathbb{P}_w(d\xi), \tag{9}$$

where $\mathbb{P}_w$ is the (in practice unknown) probability measure induced by the noise distribution, $\mathbb{1}_{X_i}(\cdot)$ is the indicator function (which returns value 1 if its argument belongs to the set X_i). Since transition probability intervals are obtained from the scenario approach theory, they contain probability $\mathrm{P}^\star$ with confidence β:

$$\mathbb{P}^V\Big\{\mathrm{P}^\star(s,a)(s')\in\mathcal{P}(s,a)(s'),\forall s\in\mathcal{S}\Big\}\geq 1-\frac{\beta}{|\mathcal{A}|\cdot|\mathcal{S}|}. \tag{10}$$

The generated iMDP has at most $|\mathcal{A}| \cdot |\mathcal{S}|$ unique probability intervals, because $\mathrm{P}^\star(s,a)(s') = \mathrm{P}^\star(s'',a)(s')$ for any $s, s'' \in \mathcal{S}$ in which a is enabled. Thus, using Boole's inequality, we have that for all probabilities $\mathrm{P}^\star(s,a)(s')$

$$\mathbb{P}^V \left\{ \mathrm{P}^\star(s,a)(s') \in \mathcal{P}(s,a)(s'), \forall s, s' \in \mathcal{S}, a \in \mathcal{A} \right\} \geq 1-\beta. \tag{11}$$

Let $\mathcal{M}^{\mathrm{P}^\star}$ denote the MDP under the true transition function $\mathrm{P}^\star$, and let $\pi \in \Pi_{\mathcal{M}^{\mathrm{P}^\star}}$ be any policy for this MDP such that a given PCTL property Φ is satisfied on the iMDP, i.e. $\mathcal{M}^{\mathrm{P}^\star} \models_\pi \Phi$. Using concepts from probabilistic simulation relations [17,28,35], it can be shown that there exists a controller F such that $(\mathcal{M}^{\mathrm{P}^\star} \models_\pi \Phi) \implies (\mathfrak{L} \models_F \Phi)$. Combining this with (11), which states that $\mathbb{P}^V\{\mathcal{M}^{\mathrm{P}^\star} \in \mathcal{M}_{\mathbb{I}}\} \geq 1-\beta$, we arrive at the condition for a β-iMDP abstraction in Def. 2.

Theorem 1 can be used to synthesise *provably correct* controllers for temporal logic specifications, but is limited to systems *without discrete dynamics*, as for MJLSs. In what follows, we will develop a framework to overcome this limitation.

4 Abstractions of Markov Jump Linear Systems

In this section, we present our main contributions to solving Problem 1. We first explain how we use the results from Sect. 3 to construct an abstraction for the continuous dynamics of an individual mode. Then, we discuss how to "combine" abstractions across discrete modes to obtain a single iMDP abstraction. Finally, we compute an optimal policy $\pi^\star$ on the obtained iMDP and show (using Theorems 2 and 3) that this policy can be refined as a controller for the hybrid dynamics.

4.1 iMDP Abstraction for Individual Modes

We construct an abstraction for each separate mode $z \in \mathcal{Z}$ of the MJLS defined by (1) using the procedure that led to Theorem 1. For simplicity, we consider rectangular partitions, but our methods are applicable for any partition into convex sets satisfying Def. 3, and even to distinct partitions across modes. We then obtain a β-iMDP abstraction $\mathcal{M}_{\mathbb{I}}^z = (\mathcal{S}, \mathcal{A}_z, s_I, \mathcal{P}_z)$ for each mode $z \in \mathcal{Z}$.

In order to reason over the *hybrid* system as a whole, we now need a sound method to "combine" the abstractions $\mathcal{M}_{\mathbb{I}}^z$ for each mode $z \in Z$ into a single abstract model. However, without careful consideration of the enabled discrete actions, the resulting model may fail to soundly abstract the overall MJLS, as different actions may be enabled in the same region of continuous states, and this would lead to spurious trajectories in the abstraction.

To exemplify this issue, consider a specific instance of Example 1, in which the room without the radiator is perfectly insulated from the other. If we naively "combine" single-mode abstractions together, then we might conclude that we will be able to heat either room to any temperature, since the two modes taken individually can heat either room. This is an example of artificial behaviour introduced in the abstraction. In reality, we can only control one room at the same time; any actions which say otherwise will not be realisable on the concrete dynamical model.

As our main contribution, we introduce in Sect. 4.2 an approach for combining single-mode iMDPs under Assumption A in a sound manner, and in Sect. 4.3 we discuss the case for Assumption B.

4.2 Abstraction Under Uncertain Markov jumps (Assumption A)

Under Assumption A, we have access to an iMDP representation $\mathcal{M}_{\mathbb{I}} = (\mathcal{Z},\mathcal{B},z_I,\mathcal{T})$ of the discrete-mode Markov jump process, which has modes in $\mathcal{Z}$, switching actions in $\mathcal{B}$, initial mode z_I, and transition probability intervals in $\mathcal{T}$. Let $\{\mathcal{M}_{\mathbb{I}}^z = (\mathcal{S},\mathcal{A}_z,s_I,\mathcal{P}_z)\}_{z\in\mathcal{Z}}$ be a set of β-iMDPs for each mode $z \in Z$, constructed as described in Sect. 4.1 with a confidence level of $\beta \in (0,1)$. We assume that these β-iMDPs have a common state space $\mathcal{S}$, and an overall action space $\mathcal{A}$. We also allow for a mode-dependent set of enabled actions; and use the notation $\mathcal{A}_z(s)$ to define actions enabled at a state s, in mode z.

To combine these modes, we use a product construction, similar to methods for constructing product automata [23]. We define our product construction among $\mathcal{M}_{\mathbb{I}}$ and $\{\mathcal{M}_{\mathbb{I}}^z\}_{z\in\mathcal{Z}}$. The *joint state/action* space of the product are the sets $\mathcal{Z}\times\mathcal{S}$ and $\mathcal{B}\times\mathcal{A}$. At a particular joint state (z,s), we define the set of enabled actions $\mathcal{A}(z,s) = \mathcal{B}(z)\times\mathcal{A}_z(s)$ as the product between the actions enabled at a particular mode, and the switches allowed in the corresponding state of the discrete iMDP. Thus, an action in the product iMDP corresponds with executing both an action in $\mathcal{A}_z$ (for the current mode z) and a discrete mode switching action in $\mathcal{B}$. The overall product iMDP under Assumption A is defined as follows:

Definition 4 (Product iMDP with mode switch control). *Let $\{\mathcal{M}_{\mathbb{I}}^z = (\mathcal{S},\mathcal{A}_z,s_I,\mathcal{P}_z)\}_{z\in\mathcal{Z}}$ be a set of β-iMDP abstractions for each mode $z \in Z$, and let $\mathcal{M}_{\mathbb{I}} = (\mathcal{Z},\mathcal{B},z_I,\mathcal{T})$ be an iMDP for the Markov jump process. Then, the product iMDP $\mathcal{M}_{\mathbb{I}}^{\times} = (S_{\times},A_{\times},s_{\times}^I,\mathcal{P}_{\times})$ is defined with*

- *Joint state space $S_{\times} = \mathcal{Z}\times\mathcal{S}$;*
- *Joint action space $\mathcal{A}_{\times} = \mathcal{B}\times\mathcal{A}$, with enabled actions $\mathcal{A}(z,s)$ in state (z,s);*
- *Initial joint state $s_{\times}^I = (z_I,s_I)$;*
- *For each $(z,s),(z',s') \in \mathcal{Z}\times\mathcal{S}$ and $(b,a) \in \mathcal{A}(z,s)$, the probability interval*

$$\mathcal{P}_{\times}\big((z,s),(b,a)\big)\big((z',s')\big) = [\underline{t}(z,b)(z')\cdot\underline{p_z}(s,a)(s'),\overline{t}(z,b)(z')\cdot\overline{p_z}(s,a)(s')]. \tag{12}$$

Here $\underline{p_z}(s,a)(s')$ and $\overline{p_z}(s,a)(s')$ are, respectively, the lower and upper bound state transition probability of β-iMDP $\mathcal{M}_{\mathbb{I}}^z$ for mode $z \in \mathcal{Z}$, and $[\underline{t}(z,b)(z'),\overline{t}(z,b)(z')]$ are the intervals in the transition function $\mathcal{T}$ of the jump process iMDP.

By construction, the product iMDP merges the individual mode abstractions and the mode-switching iMDP in a sound manner, thus avoiding the issues with spurious actions described in Sect. 4.1. The product iMDP depends on NV samples (N sets of V samples, one for each mode), hence the abstraction is a random variable on the space NV. The $\mathbb{P}^{NV}$ appearing in these theorems denotes the product measure $\mathbb{P}_{z_1}^V \otimes \mathbb{P}_{z_2}^V \cdots \mathbb{P}_{z_N}^V$ (note that the noise distribution can differ between modes). We extend Theorem 1 to the product iMDP as follows.

Theorem 2 (iMDP abstraction of controlled MJLS). *The product iMDP defined by Def. 4 is a β'-iMDP abstraction with confidence $\beta' = \beta \cdot |\mathcal{Z}|$ for the MJLS in* (1)*, which captures the mode switching iMDP $\mathcal{M}_{\mathbb{I}}$. In particular,*

$$\mathbb{P}^{NV}\Big\{(\mathcal{M}_{\mathbb{I}}^{\times} \models_{\pi} \Phi) \Longrightarrow (\mathfrak{J} \models_{F} \Phi)\Big\} \geq 1-\beta'. \tag{13}$$

We provide an outline of the proof here, whilst for a detailed proof we refer to [47, Appendix 1]. The key observation is that the product iMDP is defined as the product between $|Z|$ β-iMDPs (having intervals that are "correct" with probability at least $1-\frac{\beta}{|\mathcal{A}|\cdot|\mathcal{S}|}$, cf. (10)) and the mode switching iMDP (which is "correct" with probability one). These $|\mathcal{Z}|$ individual-mode iMDPs have $|\mathcal{A}|\cdot|\mathcal{S}|\cdot|\mathcal{Z}|$ unique intervals in total. Thus, the probability for all intervals to be correct (and thus for the product iMDP to be sound) is at least $1-\frac{\beta\cdot|\mathcal{A}|\cdot|\mathcal{S}|\cdot|\mathcal{Z}|}{|\mathcal{A}|\cdot|\mathcal{S}|} = 1-\beta'$. Finally, analogously to Theorem 1, the iMDP is a probabilistic simulation relation [28], such that the satisfaction of general PCTL formulae in the discrete abstraction guarantees the satisfaction of the same formulae in the concrete MJLS system.

4.3 Abstraction Under Unknown Markov jumps (Assumption B)

Under Assumption B, the mode transition probabilities are now completely unknown. Thus, in contrast with Sect. 4.2, we generate an abstraction that is robust to any mode we may be in.

Robustifying enabled actions First, we modify the computation of the backward reachable set in (7) to introduce a backward reachable set across all possible modes (i.e., the set that can reach d regardless of which mode we are in – note the universal quantification $\forall z \in \mathcal{Z}$ in the following equation):

$$\begin{aligned}\mathcal{G}^{-1}(d) &= \{x \in \mathbb{R}^n \,|\, d = A_z x + B_z u + q_z, u \in \mathcal{U}, \forall z \in \mathcal{Z}\} \\ &= \bigcap_{z\in\mathcal{Z}} \{x \in \mathbb{R}^n \,|\, d = A_z x + B_z u + q_z, u \in \mathcal{U}\} = \bigcap_{z\in\mathcal{Z}} \mathcal{R}_z^{-1}(d),\end{aligned} \tag{14}$$

where $\mathcal{R}_z^{-1}(d)$ is the backward reachable set for mode $z \in \mathcal{Z}$, as defined in (7).

Similar to Sect. 3, we use backward reachable set computation to define the set of enabled actions, now denoted by $\mathcal{A}_{\forall z}$, in the iMDP. Indeed, for a given partition of the state space $X = \{X_1, \ldots, X_p\}$, an action a is enabled at a state s if the backward reachable set $\mathcal{G}^{-1}(d)$ defined in (14) contains the corresponding element X_i of the partition X, i.e., a is enabled if $X_i \subseteq \mathcal{G}^{-1}(d)$. Thus, by definition, the action a is realisable on the concrete dynamical model, regardless of the current mode of operation.

Robustifying probability intervals We render the transition probability intervals robust against any mode in two steps. First, we compute the transition probability intervals $\mathcal{P}_z$ for the iMDP $\mathcal{M}_{\mathbb{I}}^z$ for each individual mode $z \in \mathcal{Z}$. For each transition (s,a,s'), the robust interval $\mathcal{P}_{\forall z}(s,a)(s')$ is then obtained as the *smallest probability interval* that contains the intervals in $\mathcal{P}_z$ for all modes $z \in \mathcal{Z}$:

$$\mathcal{P}_{\forall z}(s,a)(s') = \big[\min_{z\in\mathcal{Z}} \underline{p_z}(s,a)(s'),\ \max_{z\in\mathcal{Z}} \overline{p_z}(s,a)(s')\big], \tag{15}$$

where $\underline{p_z}(s,a)(s')$ and $\overline{p_z}(s,a)(s')$ are again the lower/upper bound probabilities of iMDP $\mathcal{M}_{\mathbb{I}}^z$. Using (15) we obtain probability intervals that are, by construction, a sound overapproximation of the probability intervals *under any mode* $z \in \mathcal{Z}$. We use this key observation to state the correctness of the resulting iMDP.

Theorem 3 (Robust iMDP with unknown mode jumps). *The robust iMDP* $\mathcal{M}_{\mathbb{I}}^{\forall z} = (\mathcal{S}, \mathcal{A}_{\forall z}, s_I, \mathcal{P}_{\forall z})$ *with actions defined through* (14) *and intervals defined by* (15) *is a* β'*-iMDP abstraction for the MJLS in* (1)*, which models state transitions robustly against any mode transition. In particular,*

$$\mathbb{P}^{NV}\Big\{(\mathcal{M}_{\mathbb{I}}^{\forall z} \models_\pi \Phi) \implies (\mathfrak{J} \models_F \Phi)\Big\} \geq 1-\beta'. \tag{16}$$

We again provide the full proof in [47, Appendix 1], while only providing an outline here. The robust iMDP is composed of intervals that contain the true transition probabilities, with a probability of at least $1-\frac{\beta}{|\mathcal{A}|\cdot|\mathcal{S}|}$. Thus, every probability interval of the robust iMDP contains the intervals for all modes $z \in \mathcal{Z}$ with probability at least $1-\frac{\beta}{|\mathcal{A}|\cdot|\mathcal{S}|}\cdot|\mathcal{Z}|$. Since the robust iMDP has $|\mathcal{S}|\cdot|\mathcal{A}|$ unique intervals, it follows that all intervals of the iMDP are correct with probability at least $1-\frac{\beta\cdot|\mathcal{A}|\cdot|\mathcal{S}|\cdot|\mathcal{Z}|}{|\mathcal{A}|\cdot|\mathcal{S}|} = 1-\beta'$. Analogous to the proof of Theorem 2, it is then straightforward to prove that this abstraction is also a β'-iMDP.

5 Synthesis for General PCTL Formulae

To synthesise optimal policies in our discrete abstraction, we use the probabilistic model checker PRISM [33]. We handle complex and nested PCTL formulae by defining a *parse tree* [7], whose leaves are atomic propositions, and whose branches are logical, temporal, or probabilistic operators. The complete formula can be verified using a bottom-up approach. As an example, consider the formula

$$\Phi = \mathsf{P}_{\geq 0.6}[\mathsf{X}\mathsf{P}_{\leq 0.5}(\neg T_C \mathsf{U}^{\leq K-1}(T_L \vee T_C))] \wedge \mathsf{P}_{\geq 0.9}[\neg T_C \mathsf{U}^{\leq K} T_G], \tag{17}$$

with atomic propositions T_C, T_L, T_G. The parse tree for this formula is shown in Fig. 2. We will explain and use this formula in the temperature control experiment in Sect. 6. When considering multiple PCTL fragments, we find a policy associated with each fragment (our example above will find two policies, one satisfying $\mathsf{P}_{\geq 0.6}[\mathsf{X}\mathsf{P}_{\leq 0.5}(\neg T_C \mathsf{U}^{\leq K-1}(T_L \vee T_C))]$ the other $\mathsf{P}_{\geq 0.9}[\neg T_C \mathsf{U}^{\leq K} T_G]$). At runtime, we choose which PCTL fragment to satisfy and apply its associated policy as $\pi^\star$.

5.1 Unsatisfied Formulae

If the PCTL formula is not satisfied by the iMDP, we refine our abstraction by increasing the number of samples used to compute the probability intervals (shown by the dashed line in Fig. 1). As also discussed in more detail and shown experimentally by [4], this refinement tightens the probability intervals, which in turn improves the probability of satisfying the property. We iteratively refine our abstraction until the

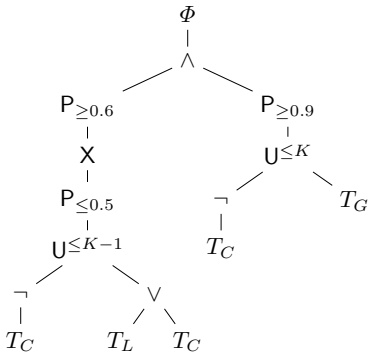

Fig. 2. Parse tree for PCTL formula (17).

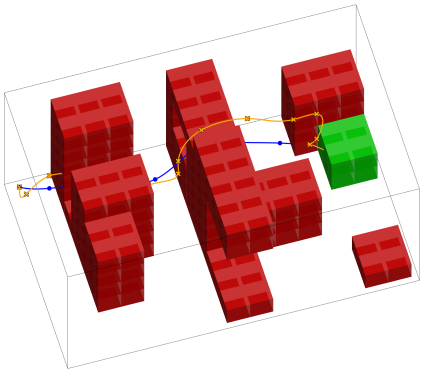

Fig. 3. Simulated paths under weak (blue) and strong (orange) wind for the drone.

formula is satisfied or until a maximum number of iterations is exceeded (which we fix a priori), in which case nothing is returned. In this way, our method is sound, but not complete: if the formula is not satisfied after the maximum number of iterations, this in general does not imply that the formula cannot be satisfied at all. However, for any policy that is returned by our algorithm, the correctness result of Theorem 3 holds.

5.2 Controller Synthesis via Policy Refinement

We refine the optimal policy $\pi^\star$ to obtain a hybrid-state feedback controller F for the MJLS, as follows. Given the current continuous state $x \in \mathcal{X}$, mode $z \in \mathcal{Z}$ and time step $k \in \mathbb{N}$, we first find the element X_i of partition X containing state x, such that $x \in X_i$. Depending on whether we consider abiding by modelling Assumption A or Assumption B, we then proceed as follows:

- For Assumption A, we find the product state $s_\times = (z,s)$ associated with the current mode $z \in \mathcal{Z}$ and state s. We then look up the optimal product action $a_\times = \pi^\star(s_\times,k) = (b,a)$ from policy $\pi^\star$, with corresponding switching action b and continuous action a.
- For Assumption B, it suffices to know state s associated with X_i only, and we directly obtain action $a = \pi^\star(s,k)$, with no switching action.

Finally, we compute the continuous control input u associated with action a by calculating the control input that drives us to the associated target point d, using $u = B_z^+(d - A_z x - q_z)$, with B_z^+ representing the pseudoinverse of B_z.

6 Numerical Experiments

We have implemented our techniques in Python, using the probabilistic model checker PRISM [33] to verify the satisfaction of PCTL formulae on iMDPs. The

codebase is available at https://github.com/lukearcus/ScenarioAbstraction. Experiments were run on a computer with 6 3.7 GHz cores and 32 GB of RAM. We demonstrate our techniques on two models: (1) a UAV motion control problem with two possible levels of noise, and (2) a building temperature regulation problem, in line with our running example from Sect. 2. Details on the UAV model and additional experimental results can be found in [47, Appendix 2].

6.1 UAV Motion Planning

We consider a more refined, *hybrid* version of the unmanned aerial vehicle (UAV) motion planning problem from [4]. We consider two discrete modes, which reflect different levels of noise, namely low and high wind speeds. We use our framework considering Assumption A. The PCTL specification $\Phi = \mathsf{P}_{\geq 0.5}[\neg O \mathsf{U}^{\leq K} G]$ requires reaching a goal set G (highlighted in green in Fig. 3), whilst avoiding obstacles O (highlighted in red). We choose a finite time horizon $K = 64$. While our theoretical contributions hold for any probability distribution for the additive noise, in this particular experiment we sample from a Gaussian.

Scalability The number of iMDP states equals the number of partitions, multiplied by the number of discrete modes, here resulting in 51,030 states. The number of transitions depends on the number of samples: with 100 samples, we generate an iMDP with 92.7 million transitions; with 200 samples, 154 million transitions. Computing the iMDP actions enabled in the abstraction is independent of sampling and takes 8.5 min; computing the transition probability intervals of the iMDP takes 70 min; formal synthesis of the optimal policy takes 40 s, and control refinement occurs online.

Variable noise affects decisions With our techniques, we synthesise a controller that accounts for different noise levels at runtime and reasons about the probability of the noise level changing. Thus, our framework makes use of the information available regarding the jump process, while at the same time reasoning explicitly over the stochastic noise affecting the continuous dynamics in each mode.

6.2 Temperature Regulation in a Building

We consider again the 2-room building temperature control problem [3] introduced in Example 1. Recall that the state $x = [T_1, T_2]^\top \in \mathbb{R}^2$ models the temperature in both rooms, and the control input (modelling the power supply to the heaters) is constrained to $u \in \mathbb{R}^2$. The values of the constants in (2) are $a_{12} = 0.022$, $b_1 = b_2 = 0.0167$, $k_f = 0.8$, $k_r = 0.4$, $x_a = 6$. The noise is distributed according to a zero-mean Gaussian with a standard deviation of 0.2.

We wish to optimise the probability of satisfying the path formula $\psi = (\neg T_C)\mathsf{U}^{\leq K}(T_G)$, with goal temperature T_G between 22 and 23°C, and critical temperature T_C less than 20°C or greater than 25°C. We partition the state space into 1600 regions, using a time horizon $K = 32$. We look into two setups, one fulfilling Assumption A (Fig. 4) and the other Assumption B (Fig. 5). We show the results for all initial continuous states, and in Fig. 4 we consider starting in mode 1 (whereas the bounds in Fig. 5 hold for any initial mode).

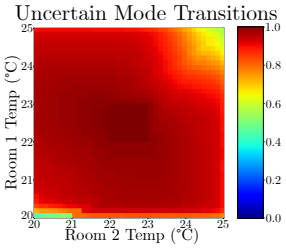

Fig. 4. Lower bound satisfaction probabilities with interval mode jump probabilities (Assumption A).

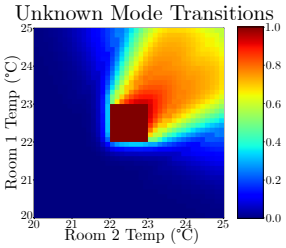

Fig. 5. Lower bound satisfaction probabilities with unknown mode jump probabilities (Assumption B).

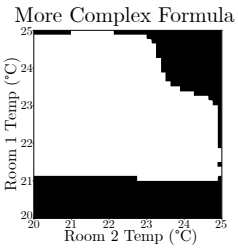

Fig. 6. States that satisfy (in white) the general PCTL formula in (17) (Assumption A).

Assumptions affect conservativism and scalability When we wish to be robust to all possible modes (cf. Assumption B), our generated iMDP is much smaller (with about 18 times fewer transitions), since we have a single robust iMDP, compared to a product iMDP. However, as expected and seen in Fig. 5, the obtained probability lower bounds are much more conservative. Thus, compared to Assumption B, Assumption A reduces the level of conservatism (because we exploit the probability intervals of the Markov jump process) at the cost of increasing the size of the abstraction.

6.3 Controller Synthesis for General PCTL Formulae

We now consider the general PCTL formula in (17) to show the applicability of our techniques beyond reach-avoid specifications. This formula requires (1) heating both rooms to a goal temperature while avoiding critical temperatures; and (2) reaching a state at the next time step, which is able to avoid entering an unwanted or critical temperature. The new atomic proposition T_L specifies that temperatures should be kept below 21°C in room 1. In Fig. 6, we show the set of iMDP states that satisfy the PCTL formula (shown in white), if the fan heater is initially in room 1 (see Example 1). Thus, we can compute a feedback controller for the MJLS satisfying the PCTL formula, unless the initial room temperature is (approximately) below 21°C, or if both initial temperatures are too high.

7 Conclusions and Future Work

We have presented a new method for synthesizing certifiably correct controllers for MJLSs with hybrid, stochastic and partly unknown dynamics. We considered both the case where an estimate of the switching probabilities across discrete operation modes is known, and the alternative instance where these probabilities are not known at all. Our experiments have demonstrated the efficacy of our methods on a number of realistic problems.

Future research directions include considering state-dependent mode switches (e.g. for models in [2,39]), estimating mode-switching probabilities with the scenario approach, and dealing with a setting where matrices are only known to belong to a convex polytope, as in [5] for non-hybrid systems.

References

1. Abate, A., D'Innocenzo, A., Benedetto, M.D.D., Sastry, S.: Markov set-chains as abstractions of stochastic hybrid systems. In: HSCC. Lecture Notes in Computer Science, vol. 4981, pp. 1–15. Springer (2008)
2. Abate, A., Katoen, J., Lygeros, J., Prandini, M.: Approximate model checking of stochastic hybrid systems. Eur. J. Control **16**(6), 624–641 (2010)
3. Abate, A., Prandini, M., Lygeros, J., Sastry, S.: Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Autom. **44**(11), 2724–2734 (2008)
4. Badings, T.S., Abate, A., Jansen, N., Parker, D., Poonawala, H.A., Stoelinga, M.: Sampling-based robust control of autonomous systems with non-gaussian noise. In: AAAI. pp. 9669–9678. AAAI Press (2022)
5. Badings, T.S., Romao, L., Abate, A., Jansen, N.: Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In: AAAI. pp. 14701–14710. AAAI Press (2023)
6. Badings, T.S., Romao, L., Abate, A., Parker, D., Poonawala, H.A., Stoelinga, M., Jansen, N.: Robust control for dynamical systems with non-gaussian noise via formal abstractions. J. Artif. Intell. Res. **76**, 341–391 (2023)
7. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
8. Belta, C., Yordanov, B., Aydin Gol, E.: Formal Methods for Discrete-Time Dynamical Systems. Springer International Publishing (2017)
9. Benbrahim, M., Kabbaj, M., Benjelloun, K.: Robust control under constraints of linear systems with markovian jumps. Int. J. Control Autom. Syst. **14**(6), 1447–1454 (2016)
10. Blackmore, L., Ono, M., Bektassov, A., Williams, B.C.: A probabilistic particle-control approximation of chance-constrained stochastic predictive control. IEEE Trans. Robotics **26**(3), 502–517 (2010)
11. Boucheron, S., Lugosi, G., Massart, P.: Concentration Inequalities - A Nonasymptotic Theory of Independence. Oxford University Press (2013)
12. Boukas, E.K., Benzaouia, A.: Stability of discrete-time linear systems with markovian jumping parameters and constrained control. IEEE Trans. Autom. Control. **47**(3), 516–521 (2002)
13. Cai, H., Li, P., Su, C., Cao, J.: Robust model predictive control for a class of discrete-time markovian jump linear systems with operation mode disordering. IEEE Access **7**, 10415–10427 (2019)
14. Campi, M.C., Garatti, S.: The exact feasibility of randomized solutions of uncertain convex programs. SIAM J. Optim. **19**(3), 1211–1230 (2008)
15. Campi, M.C., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. J. Optim. Theory Appl. **148**(2), 257–280 (2011)
16. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. Annu. Rev. Control. **33**(2), 149–157 (2009)
17. Cauchi, N., Laurenti, L., Lahijanian, M., Abate, A., Kwiatkowska, M., Cardelli, L.: Efficiency through uncertainty: scalable formal synthesis for stochastic hybrid systems. In: HSCC. pp. 240–251. ACM (2019)
18. Clarke, E.M., Fehnker, A., Han, Z., Krogh, B.H., Stursberg, O., Theobald, M.: Verification of hybrid systems based on counterexample-guided abstraction refinement. In: TACAS. Lecture Notes in Computer Science, vol. 2619, pp. 192–207. Springer (2003)
19. Cunha, R.F., Gabriel, G.W., Geromel, J.C.: Robust partial sampled-data state feedback control of markov jump linear systems. Int. J. Syst. Sci. **50**(11), 2142–2152 (2019)

20. Do Costa, O.L.V., Marques, R.P., Fragoso, M.D.: Discrete-Time Markov Jump Linear Systems. Springer (2005)
21. de Farias, D.P., Geromel, J.C., do Val, J.B.R., Costa, O.L.V.: Output feedback control of markov jump linear systems in continuous-time. IEEE Trans. Autom. Control. **45**(5), 944–949 (2000)
22. Gabriel, G.W., Geromel, J.C.: Performance evaluation of sampled-data control of markov jump linear systems. Autom. **86**, 212–215 (2017)
23. Gécseg, F.: Products of Automata, EATCS Monographs on Theoretical Computer Science, vol. 7. Springer (1986)
24. Givan, R., Leach, S.M., Dean, T.L.: Bounded-parameter markov decision processes. Artif. Intell. **122**(1-2), 71–109 (2000)
25. González-Trejo, J.I., Hernández-Lerma, O., Reyes, L.F.H.: Minimax control of discrete-time stochastic systems. SIAM J. Control. Optim. **41**(5), 1626–1659 (2002)
26. Hahn, E.M., Han, T., Zhang, L.: Synthesis for PCTL in parametric markov decision processes. In: NASA Formal Methods. Lecture Notes in Computer Science, vol. 6617, pp. 146–161. Springer (2011)
27. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects Comput. **6**(5), 512–535 (1994)
28. Hermanns, H., Parma, A., Segala, R., Wachter, B., Zhang, L.: Probabilistic logical characterization. Inf. Comput. **209**(2), 154–172 (2011)
29. Hespanha, J.P., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proc. IEEE **95**(1), 138–162 (2007)
30. Hu, L., Shi, P., Frank, P.M.: Robust sampled-data control for markovian jump linear systems. Autom. **42**(11), 2025–2030 (2006)
31. Jiang, B., Wu, Z., Karimi, H.R.: A traverse algorithm approach to stochastic stability analysis of markovian jump systems with unknown and uncertain transition rates. Appl. Math. Comput. **422**, 126968 (2022)
32. Knight, J.C.: Safety critical systems: challenges and directions. In: ICSE. pp. 547–550. ACM (2002)
33. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer (2011)
34. Lahijanian, M., Andersson, S.B., Belta, C.: Formal verification and synthesis for discrete-time stochastic systems. IEEE Trans. Autom. Control. **60**(8), 2031–2045 (2015)
35. Lavaei, A., Soudjani, S., Abate, A., Zamani, M.: Automated verification and synthesis of stochastic hybrid systems: A survey. Autom. **146**, 110617 (2022)
36. Li, W., Xu, Y., Li, H.: Robust l_2-l_∞ filtering for discrete-time markovian jump linear systems with multiple sensor faults, uncertain transition probabilities and time-varying delays. IET Signal Process. **7**(8), 710–719 (2013)
37. Lindemann, L., Hu, H., Robey, A., Zhang, H., Dimarogonas, D.V., Tu, S., Matni, N.: Learning hybrid control barrier functions from data. In: CoRL. Proceedings of Machine Learning Research, vol. 155, pp. 1351–1370. PMLR (2020)
38. Lun, Y.Z., Wheatley, J., D'Innocenzo, A., Abate, A.: Approximate abstractions of markov chains with interval decision processes. In: ADHS. IFAC-PapersOnLine, vol. 51, pp. 91–96. Elsevier (2018)
39. Lunze, J., Lamnabhi-Lagarrigue, F. (eds.): Handbook of Hybrid Systems Control: Theory, Tools, Applications. Cambridge University Press (2009)
40. Mazo, Jr., M., Davitian, A., Tabuada, P.: PESSOA: A tool for embedded controller synthesis. In: CAV. Lecture Notes in Computer Science, vol. 6174, pp. 566–569. Springer (2010)

41. Moggi, E., Farjudian, A., Duracz, A., Taha, W.: Safe & robust reachability analysis of hybrid systems. Theor. Comput. Sci. **747**, 75–99 (2018)
42. Morais, C.F., Palma, J.M., Peres, P.L.D., Oliveira, R.C.L.F.: An LMI approach for H2 and H∞ reduced-order filtering of uncertain discrete-time markov and bernoulli jump linear systems. Autom. **95**, 463–471 (2018)
43. Nejati, A., Soudjani, S., Zamani, M.: Compositional construction of control barrier functions for continuous-time stochastic hybrid systems. Autom. **145**, 110513 (2022)
44. Platzer, A.: Logics of dynamical systems. In: LICS. pp. 13–24. IEEE Computer Society (2012)
45. Puggelli, A., Li, W., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Polynomial-time verification of PCTL properties of mdps with convex uncertainties. In: CAV. Lecture Notes in Computer Science, vol. 8044, pp. 527–542. Springer (2013)
46. Ramponi, F., Chatterjee, D., Summers, S., Lygeros, J.: On the connections between PCTL and dynamic programming. In: HSCC. pp. 253–262. ACM (2010)
47. Rickard, L., Badings, T.S., Romao, L., Abate, A.: Controller synthesis for markov jump linear systems with non-gaussian noise. Tech. rep. (2023), https://www.lukerickard.co.uk/RBRA23.pdf
48. Robey, A., Lindemann, L., Tu, S., Matni, N.: Learning robust hybrid control barrier functions for uncertain systems. In: ADHS. IFAC-PapersOnLine, vol. 54, pp. 1–6. Elsevier (2021)
49. Romao, L., Papachristodoulou, A., Margellos, K.: On the exact feasibility of convex scenario programs with discarded constraints. IEEE Trans. Autom. Control. **68**(4), 1986–2001 (2023)
50. Soudjani, S.E.Z., Abate, A.: Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. SIAM J. Appl. Dyn. Syst. **12**(2), 921–956 (2013)
51. Tabuada, P.: Verification and Control of Hybrid Systems - A Symbolic Approach. Springer (2009)
52. Tian, E., Yue, D., Wei, G.: Robust control for markovian jump systems with partially known transition probabilities and nonlinearities. J. Frankl. Inst. **350**(8), 2069–2083 (2013)
53. Tkachev, I., Abate, A.: Characterization and computation of infinite horizon specifications over markov processes. Theoretical Computer Science **515**, 1–18 (2014)
54. do Valle Costa, O.L., Fragoso, M.D.: Discrete-time lq-optimal control problems for infinite markov jump parameter systems. IEEE Trans. Autom. Control. **40**(12), 2076–2088 (1995)
55. do Valle Costa, O.L., Fragoso, M.D., Todorov, M.G.: A detector-based approach for the h_2 control of markov jump linear systems with partial information. IEEE Trans. Autom. Control. **60**(5), 1219–1234 (2015)
56. Zhang, L., Boukas, E.K.: Stability and stabilization of markovian jump linear systems with partly unknown transition probabilities. Autom. **45**(2), 463–468 (2009)