Proceedings of Machine Learning Research vol XX:1–22, 2023

Formal Controller Synthesis for Markov Jump Linear Systems with Uncertain Dynamics

Luke Rickard[1] RICKARD@ROBOTS.OX.AC.UK
Thom Badings[2] THOM.BADINGS@RU.NL
Licio Romao[1] LICIO.ROMAO@CS.OX.AC.UK
Nils Jansen[2] N.JANSEN@SCIENCE.RU.NL
Alessandro Abate[1] ALESSANDRO.ABATE@CS.OX.AC.UK
[1] *Department of Computer Science, University of Oxford, Oxford, UK*
[2] *Department of Software Science, Radboud University, Nijmegen, the Netherlands*

Abstract

Automated synthesis of provably correct controllers for cyber-physical systems is crucial for deploying these systems in safety-critical scenarios. However, their hybrid features and stochastic or unknown behaviours make this synthesis problem challenging. In this paper, we propose a method for synthesizing controllers for Markov jump linear systems (MJLSs), a particular class of cyber-physical systems, that certifiably satisfy a requirement expressed as a specification in probabilistic computation tree logic (PCTL). An MJLS consists of a finite set of linear dynamics with unknown additive disturbances, where jumps between these modes are governed by a Markov decision process (MDP). We consider both the case where the transition function of this MDP is given by probability intervals or where it is completely unknown. Our approach is based on generating a finite-state abstraction which captures both the discrete and the continuous behaviour of the original system. We formalise such abstraction as an interval Markov decision process (iMDP): intervals of transition probabilities are computed using sampling techniques from the so-called "scenario approach", resulting in a probabilistically sound approximation of the MJLS. This iMDP abstracts both the jump dynamics between modes, as well as the continuous dynamics within the modes. To demonstrate the efficacy of our technique, we apply our method to multiple realistic benchmark problems, in particular, temperature control, and aerial vehicle delivery problems.

Keywords: Robust control, Markov jump linear systems, uncertain models, safety guarantees

1. Introduction

In a world where autonomous cyber-physical systems are becoming more prevalent, it is important to develop methods for the safe control of these systems (Knight, 2002). Cyber-physical systems are characterised by coupling of digital-computation elements with physical dynamical components. The result is a system with *hybrid* (namely, discrete/continuous) features, whereby continuous dynamics may "jump" between different discrete modes of operation (Lavaei et al., 2021). Since cyber-physical systems are increasingly deployed in safety-critical settings, ensuring that these systems meet certain formal specifications is an important goal.

Formal controller synthesis. Often, complex and rich specifications cannot be expressed as traditional control-theoretic objectives, which by and large relate to stability and convergence requirements. These requirements can be expressed in temporal logics, which provide a rich language for specifying the desired behaviour of dynamical systems (Platzer, 2012). Probabilistic computation tree logic (PCTL, Hansson and Jonsson 1994), is widely used to define temporal requirements on the probabilistic behaviour of probabilistic systems, and in particular also for classes of cyber-physical systems. For example, we can use PCTL to specify that, in the context of a temperature control problem in a building, the room temperature must be controlled to 21°C after 100 seconds with at least a 75% chance. Leveraging probabilistic verification tools (Baier and Katoen, 2008), we can then synthesise a controller that ensures the satisfaction of such a PCTL formula (Hahn et al., 2011).

Markov jump linear systems. Markov jump linear systems (MJLSs) (do Costa et al., 2005) are a well known class of stochastic hybrid models suitable for capturing the behaviour of cyber-physical systems (Lavaei et al., 2021). An MJLS consists of a finite set of (possibly probabilistic) linear dynamics (called operation "modes"), where jumps between these modes are governed by a Markov chain (MC) or (if switchings among modes are controllable) by a Markov decision process (MDP). Despite having linear dynamics at each mode, the overall dynamics are non-linear due to the switching behaviour. MJLSs have, for example, been used to model networked control systems, where the different modes relate to package loss (Hespanha et al., 2007; Morais et al., 2018).

Due to their wide applicability and relevance, several discrete and continuous time MJLS control problems have been studied, such as stability (Boukas and Yang, 1995; Zhang and Boukas, 2009), H_∞-controller design (de Farias et al., 2000; do Valle Costa et al., 2015; Gabriel and Geromel, 2017; Cunha et al., 2019), and optimal control (do Valle Costa and Fragoso, 1995; Hu et al., 2006). In this paper, we consider a behaviourally rich class of discrete-time MJLSs, for which the continuous dynamics (in each mode) is affected by an additive stochastic process noise. This noise may arise from inaccurate modelling of the dynamics or from the presence of aleatoric components, e.g., wind gusts affecting the dynamics of a drone (Blackmore et al., 2010). We now highlight two factors of uncertainty in MJLSs that we specifically consider in this work.

1) Unknown noise distribution. Similar to Blackmore et al. (2010), we assume the distribution of the noise affecting the continuous dynamics to be arbitrary and unknown, thus potentially non-Gaussian, unlike what is commonly assumed in the literature (Park et al., 2013). We assume access to a simulator or to historical data, which provides us samples (observations) of the noise, allowing us to provide probably approximately correct (PAC) guarantees on the behaviour of the MJLS.

2) Uncertainty in the Markov jumps. In a similar spirit as in Morais et al. (2018), we assume that the transition probabilities of the Markov jump process are not precisely known. However, unlike Morais et al. (2018), we consider two different semantics for this uncertainty: (1) transition probabilities between modes are given by intervals; or (2) these probabilities are not known at all (Jiang et al., 2022; Li et al., 2013). More details on the considered model are presented in Sect. 2.

Problem statement. Given an MJLS subject to uncertainty deriving from 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.

Finite-state abstractions. We develop a technique for abstracting Markov jump linear systems by leveraging methods introduced in Badings et al. (2022a) for linear, time-invariant models. If the transition probabilities of the Markov jump process are given as intervals, we capture both the continuous-state dynamics and the discrete-mode jumps within a single abstract interval MDP model, which is an MDP whose transition probabilities lie within intervals (formal semantics will be discussed below and are in Cauchi et al. (2019)). If, instead, the transition function of the Markov jump MDP is unknown, we generate iMDP abstractions only for the continuous-state dynamics, and enforce instead robustness against all possible discrete-mode changes. In line with Badings et al. (2022a), we rely on sampling techniques from the *scenario approach* (Campi et al., 2009) to compute intervals that we associate to the transition probabilities for the abstract MDP model.

Once an iMDP abstraction is obtained, the state-of-the-art verification tool PRISM (Kwiatkowska et al., 2011) formally synthesises a policy that satisfies the given specification *under all realizations* of the transition probabilities within their corresponding intervals. We show that the iMDP soundly abstracts the original Markov jump system with a user-specified confidence probability, such that probabilistic guarantees carry over to the original model with the same probability.

Summary of contributions. We propose a controller synthesis framework applicable to temporal requirements expressed as PCTL formulae and for a broad class of hybrid modes, namely MJLSs, thus further developing the techniques in (Badings et al., 2022a) for (non-hybrid) dynamical systems. Our main contribution is a method to generate sound abstractions as interval MDPs, capturing both the continuous and discrete dynamics of a MJLS. Also, contrary to standard literature on MJLSs, we drop the assumption that the noise distributions affecting the continuous dynamics and the transition probabilities of the Markov jump process are precisely known.

Related Work

Techniques for providing safety guarantees for MJLSs can largely be split into two approaches, respectively being *abstraction-free* and *abstraction-based*.

Abstraction-free methods derive safety guarantees without the need to create simpler abstract models. One such technique is the design of barrier functions (Lindemann et al., 2020; Nejati et al., 2022; Robey et al., 2021), allowing one to define a set of safe control inputs that keep the system within safe states. Another approach is that of (probabilistic) reachability set computation, where the goal is to evaluate if the system will reach a certain state over a given time horizon. In Abate et al. (2008b); Moggi et al. (2018) reachability set computation is used for (stochastic) hybrid systems.

Alternatively, abstraction-based methods (Tabuada, 2009) analyse a simpler model of the system and transfer the obtained results (safety guarantees, or synthesised policies) back to the original system. Various approaches exist for creating abstractions of different forms, including counterexample-guided abstractions (Clarke et al., 2003), abstractions as timed automata (Ratschan and She, 2005), and abstractions as Markov models (Abate et al., 2008a).

Related to the approaches detailed above, in robust control 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 Benbrahim et al. (2016); Cai et al. (2019); Tian et al. (2013).

2. Background

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 $P \colon \mathcal{S} \times \mathcal{A} \rightharpoonup \mathit{Dist}(\mathcal{S})$ is a (partial) probabilistic transition function, with $\mathit{Dist}(\mathcal{S})$ the set of all probability distributions over $\mathcal{S}$.

Given a state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the next state is sampled from $P(s, a)$. We call a tuple (s, a, s') with probability $P(s, a)(s') > 0$ a *transition*. A Markov chain (MC) is an MDP with trivially a single action allowed at each state. We consider deterministic (or pure) policies, $\pi : S^* \to \mathcal{A}$, which map a finite sequence S^* of states to actions (this encompasses the special case of time-dependent policies). The set of all possible policies for $\mathcal{M}$ is denoted by $\Pi_{\mathcal{M}}$. Since we wish to abstract continuous dynamics, it is useful to define a labelling function $L : \mathbb{R}^n \to \mathcal{S}$. A definition of this labelling function is given in Sect. 3.1.

2.2. Interval Markov Decision Processes

Interval Markov decision processes (iMDPs) extend regular MDPs with uncertain transition probabilities. 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 an MDP, whereas $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightharpoonup 2^{Dist(\mathcal{S})}$ maps states and actions to a subset of distributions over successor states. With each particular transition (s, a, s') defining $\mathcal{P}(s, a)(s')$ which maps to *intervals* of the form $\mathbb{I} = \{[a, b] \mid a, b \in (0, 1], a \leq b\}$.

Intuitively, an iMDP encompasses a set of MDPs differing only in their transition probabilities: at each state and given an action, any of them can be elicited, resulting in a corresponding probability distribution over next states. For brevity, we denote by $P \in \mathcal{P}$ an assignment for the transition probabilities of an MDP within this set, such that $P(s, a)(s') \in \mathcal{P}(s, a)(s') \,\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$, and $P(s, a) \in Dist(\mathcal{S})$. The resulting MDP is denoted by $\mathcal{M}_{\mathbb{I}}[P]$. For iMDPs, we are often interested in synthesizing a policy $\pi \in \Pi_{\mathcal{M}_{\mathbb{I}}}$ that maximises the probability $\mathsf{P}^{\pi}(\mathcal{M}_{\mathbb{I}}[P] \models \psi)$ of satisfying a specification ψ (formally defined in Sect. 2.4) under $\mathcal{M}_{\mathbb{I}}[P]$ for the *worst-case* assignment $P \in \mathcal{P}$ (which is determined by a so-called *adversary*). In other words, an optimal iMDP policy π^* satisfies

$$\pi^* = \underset{\pi \in \Pi_{\mathcal{M}_{\mathbb{I}}}}{\arg\max} \min_{P \in \mathcal{P}} \mathsf{P}^{\pi}(\mathcal{M}_{\mathbb{I}}[P] \models \psi). \tag{1}$$

It has been shown in Puggelli et al. (2013); Lun et al. (2018) that, much like for MDPs (Puterman, 1994), deterministic policies suffice to obtain optimal values for (i)MDPs. For an MDP, there is no nondeterminism in the transition function P and the minimisation in Eq. (1) vanishes.

2.3. Markov Jump Linear Systems

Let $\mathcal{N} = \{1, \ldots, N\}$ be an index set representing the discrete modes of a Markov jump linear system, and consider the collections of real-valued matrices $A = (A_1, \ldots, A_N)$, $A_i \in \mathbb{R}^{n \times n}$, and $B = (B_1, \ldots, B_N)$, $B_i \in \mathbb{R}^{n \times m}$; and of vectors $q = (q_1, \ldots, q_N), q_i \in \mathbb{R}^n$. Each index $i \in \mathcal{N}$ defines a linear system, with state space of dimension n and control space of dimension m.

Jumps between modes are governed by an MDP $\mathcal{M}$ with a finite set of states $\mathcal{N}$ (the discrete modes), a finite set of actions $\mathcal{L} = \{1, \ldots, M\}$ for some $M \in \mathbb{N}$ (without loss of generality, we assume that all actions are enabled at each state), and transition probability function $\hat{P} \colon \mathcal{N} \times \mathcal{L} \rightharpoonup Dist(\mathcal{N})$. We denote the (bounded) continuous state at time k by $x(k) \in \mathcal{X} \subseteq \mathbb{R}^n$, and the discrete mode by $r(k) \in \mathcal{N}$. Then, given initial state $x(0), r(0)$, the hybrid state of the MJLS is updated as

$$\begin{cases} x(k+1) = A_{r(k)} x(k) + B_{r(k)} u(k) + q_{r(k)} + w_{r(k)}(k) \\ r(k+1) \sim \hat{P}(r(k), l(k)), \end{cases} \tag{2}$$

where $u(k) \in \mathcal{U} \subseteq \mathbb{R}^m$ is the constrained control input to the continuous dynamics, and $l(k) \in \mathcal{L}$ is the discrete action. The continuous dynamics are affected by some process noise $w_{r(k)}(k) \in \Delta_{r(k)}$, which may follow an entirely unknown distribution. Note that the noise $w_{r(k)}(k)$ vector can vary across the discrete modes of the overall jump model. Important for our setting, the inputs $u(k)$ and $l(k)$ are *jointly determined* by a feedback controller Ω of the following form:

Definition 1 *A (time-varying) feedback* controller $\Omega \colon \mathcal{X} \times \mathcal{N} \times \mathbb{N} \to \mathcal{U} \times \mathcal{L}$ *for the Markov jump linear system is a function that maps a continuous state* $x(k) \in \mathcal{X}$*, a discrete mode* $r(k) \in \mathcal{N}$*, and a time step* $k \in \mathbb{N}$ *to a continuous control input* $u(k) \in \mathcal{U}$ *and a discrete mode action* $l(k) \in \mathcal{L}$*.*

Example 1 *Consider a temperature regulation problem, similar to that in Abate et al. (2008b). We have two portable heaters, namely a fan heater and a radiator, to heat two rooms of a building. We define two modes* $\mathcal{N} = \{1, 2\}$*, relating to the fan heater being in room 1 or 2 respectively (and the radiator in the other room): swapping the heaters between rooms is modelled by an MDP, where we account for some probability that this mode switching action fails. The state* $x \in \mathbb{R}^n$ *models both room temperatures, and the power of both heaters can be adjusted in the range* $u \in [0, 1]$ *(being fully on and off). The room temperature dynamics can be naturally modelled using an MJLS.*

2.4. Probabilistic Computation Tree Logic

Probabilistic computation tree logic (PCTL) uses the following syntax (Baier and Katoen, 2008):

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

Here, $\sim \in \{<, \leq, \geq, >\}$ is a comparison operator and λ is a probability threshold, and Φ defines state formulae, whereas ψ are path formulae. Informally, the syntax consists of state labels $p \in AP$ in a set of atomic propositions AP, Boolean connectors for 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 must satisfy a path formula ψ with probability exceeding (or below) some given threshold λ. Recall from Eq. (1) that for iMDPs $\mathcal{M}_{\mathbb{I}}$, this threshold must hold under the *worst-case realization* of the interval probabilities $P \in \mathcal{P}$, in which case we write $\mathsf{P}_{\sim\lambda}(\mathcal{M}_{\mathbb{I}}[P] \models \psi)$.

The satisfaction relation $s \models \Phi$ defines whether a PCTL formula Φ holds true in a state s of a given model, or whether ψ holds true in a path $s_0 s_1 s_2 \ldots$. Formal definitions for semantics and model checking are provided in Hansson and Jonsson (1994); Baier and Katoen (2008).

Through a slight abuse of notation, we also say that a state x, in mode r, satisfies Φ if the associated iMDP state $s = (L(x), r)$ satisfies Φ. For optimal policies, as in Eq. (1), we consider a PCTL path formula ψ, and optimise the probability to satisfy it. The type of optimisation (max vs min) depends on the requirement $\sim \in \{<, \leq, \geq, >\}$ in the probabilistic operator.

2.5. Problem Statement

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

Problem 1 *Consider the MJLS given in* (2) *and let* Φ *be a PCTL formula. Our goal is to define a control policy* $\Omega \colon \mathcal{X} \times \mathcal{N} \times \mathbb{N} \to \mathcal{U} \times \mathcal{L}$*, such that* $(x(0), r(0)) \models \Phi$*.*

In particular, we solve this problem under two sets of assumptions for the Markov jump process.

Assumption A (Uncertain Markov jumps) *Each transition probability of the MDP $\mathcal{M}$ driving the Markov jump process is known up to a certain interval, i.e., only an iMDP representation of the jump process is known.*

Assumption B (Unknown Markov jumps) *The Markov jumps are driven by a Markov chain (MC), thus notably not an MDP,*[1] *for which we can measure the current mode, but the transition function (and hence the graph structure) is unknown.*

For Assumption A, we will exploit the transition function to synthesise a controller while reasoning over the joint probability distribution over the state $x(k)$ and mode $r(k)$. However, under Assumption B, we do not know the distribution over successor modes $r(k+1)$, so we cannot reason over the joint distribution - instead, our goal will be to be *robust* against any mode changes that may occur.

3. Abstractions of Markov Jump Linear Systems

In this section, we describe our approach for abstracting MJLSs. We first introduce an abstraction for the continuous dynamics of one individual mode. Then, we discuss how to “stitch” together abstractions across discrete mode jumps to obtain a single iMDP abstraction. We then compute an optimal policy π^* on this iMDP using Eq. (1), which (due to the soundness of the abstraction, see Theorem 4) can be converted into a certifiably correct controller for the MJLS. We provide an overview of our approach in Fig. 1.

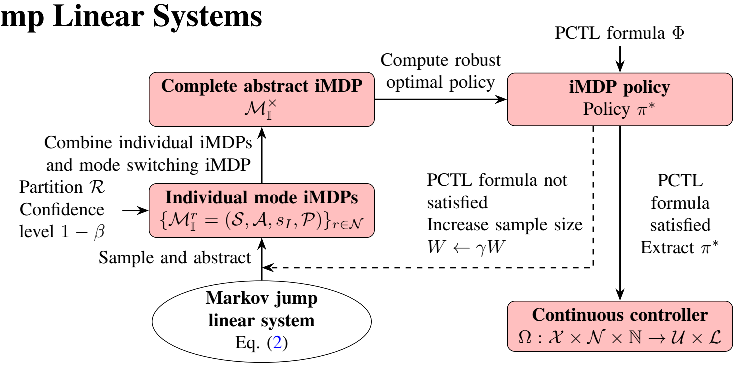
PCTL formula Φ
Compute robust optimal policy
Complete abstract iMDP $\mathcal{M}_{\mathbb{I}}^{\times}$
iMDP policy Policy π^*
Combine individual iMDPs and mode switching iMDP
Partition $\mathcal{R}$ Confidence level $1-\beta$
Individual mode iMDPs $\{\mathcal{M}_{\mathbb{I}}^{r} = (\mathcal{S}, \mathcal{A}, s_I, \mathcal{P})\}_{r \in \mathcal{N}}$
PCTL formula not satisfied Increase sample size $W \leftarrow \gamma W$
PCTL formula satisfied Extract π^*
Sample and abstract
Markov jump linear system Eq. (2)
Continuous controller $\Omega : \mathcal{X} \times \mathcal{N} \times \mathbb{N} \to \mathcal{U} \times \mathcal{L}$

Figure 1: An overview of our approach. We start with an MJLS and output a controller for that MJLS.

3.1. Abstract iMDPs for Individual Modes

Formal abstractions of linear time-invariant (LTI) systems have been studied in Badings et al. (2022a,b). Since each mode of an MJLS has linear dynamics, these techniques can be applied directly to build an iMDP abstraction separately for each mode, of which we summarise the key components below. Details on this abstraction, and, in particular, details on the sampling-based approach to generate probability intervals, are available in Appendix A.

States. The continuous state space $\mathcal{X} \subseteq \mathbb{R}^n$ is discretised into a partition $\mathcal{R} = \{R_1, \ldots, R_p\}$ of $|\mathcal{R}|$ convex polytopes, with an additional absorbing region R_a representing all $x \notin \mathcal{X}$. The function $L\colon \mathbb{R}^n \to \{1, 2, \ldots, |\mathcal{R}|+1\}$ maps continuous states to discrete regions, and the inverse mapping is denoted by $R_i = L^{-1}(i) \subset \mathcal{X}$. We then define a set of states, $\mathcal{S} = \{s_1, \ldots, s_{p+1}\}$, where each state is associated with a region in the partition, and one state is associated with the absorbing region.

[1] Since the transition function is unknown, reasoning about MDP actions is not informative and any MDP should be reduced to an MC by fixing an arbitrary policy.

Actions. We define a set of actions, $\mathcal{A} = \{a_1, \dots, a_q\}$, in the iMDP using a finite set of target points $d_1, \dots, d_q \in \mathcal{X}$. Each action in the iMDP is associated with attempting to drive the noiseless successor state to the target point. In order to ensure actions are associated with a valid control input, an action is only enabled in a state of the abstraction if any continuous state within the element of the partition corresponding to the discrete state can be steered to the associated target point. In other words, this is equivalent as saying that the element of the partition associated with a discrete state is contained in the backward reachable set of the target point, defined as

$$\mathcal{G}_i(d_j) = \{x \in \mathbb{R}^n \mid d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\}, i \in \mathcal{N}. \tag{4}$$

Transition probabilities. The transition probability intervals $\mathcal{P}(s, a, s')$ are learned through a sampling-based approach from a finite set of W i.i.d. observations of the noise at each target point. This method does not require the noise distribution to be known. The result is a set of intervals on the transition probabilities for each state-action pair, with a guarantee $\mathbb{P}^W\{P(s,a)(s') \in \mathcal{P}(s,a)(s')\} \geq 1-\beta$ that the true probability $P(s,a)(s')$ lies in this interval with at least confidence $1-\beta$.

3.2. Extension: Abstractions of Markov Jump Linear Systems

When considering a hybrid system comprising discrete jumps, a naive approach is to extend Badings et al. (2022a) by "merging" the abstractions derived for each of the individual discrete mode. However, without careful consideration of the enabled actions in each discrete mode, the resulting model will fail to soundly abstract the overall MJLS.

To exemplify this issue, consider a specific instance of Example 1, in which the room without the radiator is entirely insulated, such that we cannot affect its temperature: if we naively stitch single-mode abstractions together, then we might conclude that we will be able to heat either room to any temperature. Indeed, since the two modes taken individually can heat either room, so together they must be able to heat both rooms. This is evidently not a sound abstraction. As soon as we uncover the true mode we are in, we will only be able to control one room, any actions which say otherwise will be unable to be implemented in reality.

3.3. Controlled Jumps Between Discrete Modes

We first investigate how to solve Problem 1 under Assumption A, whereby we have access to an iMDP representation of the Markov jump process. In this case, we wish to generate an abstraction that models both the continuous dynamics and the discrete mode transitions. We achieve this by taking a product of the individual abstract iMDPs with the iMDP governing the mode jumps.

Abstractions of MJLSs with controlled mode switching. First, we define the states and actions as the Cartesian product between the individual iMDP state/actions and the mode switching iMDP state/actions. As such, we obtain an abstraction whose states (actions) are tuples of individual iMDP and discrete mode states (actions). From the action space, we again only enable a subset of actions, as in Sect. 3.1. In this case, for each state, we only enable actions in that state if they were enabled in the same continuous state in the iMDP describing the dynamics of the mode in that state.

We then use an iMDP to model our (partial) knowledge of the Markov jumps, allowing for an estimate of the mode switches without needing to know these precisely. While estimating these intervals is out of the scope of this paper, we remark this could be achieved with a similar sampling-based approach as that used to calculate the abstraction probabilities.

Definition 2 (Product of iMDPs with controlled mode switching) *Let $\mathcal{M}_{\mathbb{I}} = (\mathcal{N}, \mathcal{L}, r_I, \mathcal{P})$ be an iMDP defining the jumps between modes, and then let $\{\mathcal{M}_{\mathbb{I}}^i = (\mathcal{S}, \mathcal{A}, s_I, \mathcal{P}_i)\}_{i \in \mathcal{N}}$ be a set of iMDPs defining the state transitions within the modes, with a common state and action space $\mathcal{S}, \mathcal{A}$. We define the joint state space as the Cartesian product $S_\times = \mathcal{N} \times \mathcal{S}$. We then define a joint action space $A_\times = \mathcal{L} \times \mathcal{A}$ (of which only a state-dependent subset is enabled in each state). Then we can define the joint interval probabilities as*

$$\mathcal{P}_\times((r,s)(l,a))((r',s')) = [\underline{P}(r,l)(r') \cdot \underline{P_r}(s,a)(s'),\ \overline{P}(r,l)(r') \cdot \overline{P_r}(s,a)(s')], \tag{5}$$

where $\overline{P_r}(s,a)(s')$ refers to the upper bound state transition probability within the current mode r, and $\underline{P_r}(s,a)(s')$ refers to the lower bound. The combined iMDP is $\mathcal{M}_{\mathbb{I}}^{\times} = (S_\times, A_\times, s_\times, \mathcal{P}_\times)$.

3.4. Abstractions for Unknown Markov Jumps

We now consider Assumption B, being careful to avoid enabling impossible actions.

States in the combined iMDP are defined identically to the individual iMDPs for each mode. In order to properly define the actions, we modify the computation of the backward reachable set in Eq. (4) to introduce a backward reachable set across all possible modes (that is, this is the set of points that can reach d_j regardless of which mode we are in), as follows:

$$\mathcal{D}(d_j) = \{x \in \mathbb{R}^n \mid d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, \forall i \in \mathcal{N}\}. \tag{6}$$

To calculate this backward reachable set, we use the following theorem, which relates the combined backward reachable set with the individual backward reachable sets of the various modes.

Theorem 3 (Backwards reachable set of the MJLS dynamics) *The set of states that can reach a target state d_j in any of the dynamics corresponding to the discrete modes, is equivalent to the intersection of the backward reachable sets associated to those dynamics:*

$$\{x \in \mathbb{R}^n | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, \forall i \in \mathcal{N}\} = \bigcap_{i \in \mathcal{N}} \{x \in \mathbb{R}^n | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\}. \tag{7}$$

The proof of Theorem 3 is provided in Appendix B. Crucially, note that a state that is fully contained within this backward reachable set intersection $\mathcal{D}(d_j)$ will have the action to d_j enabled, *regardless of the discrete mode*. Thus, we can employ this backward reachable set intersection to generate an abstraction that is robust against the discrete mode of the jump process.

In practice, we obtain the intersection in Eq. (7) used to calculate enabled actions in two steps. First, for each individual discrete mode, we calculate the set of enabled actions in each state under its linear dynamics. Then, for each state, we take the intersection of the enabled actions across all the modes, resulting in a set of actions that is enabled in that state *under any discrete mode*.

Finally, to ensure the true transition probability of the MJLS is contained within the interval, we generate an interval containing the intervals of every individual iMDP. This necessarily provides a conservative estimate of the transition probability, since we will later always consider the worst case within the interval. We can thus state the following.

Theorem 4 (Sound abstraction of MJLS) *By definition of the enabled actions and transition probabilities as above, the generated iMDP is a sound abstraction of the MJLS.*

The proof of Theorem 4 is provided in Appendix B. Note that our overall method has been proven to be sound, but it is not complete: finding a controller guarantees satisfaction of the specification, but inability to find one does not prove the contrary.

4. Numerical Experiments

We have implemented our techniques in Python, using a modified version of PRISM (Kwiatkowska et al., 2011) to compute policies for iMDPs and verify satisfaction of the formula. The codebase is available at https://github.com/lukearcus/ScenarioAbstraction. All 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. Further details on the models used in these examples, and additional results, can be found in Appendix C.

4.1. UAV Motion Planning

We first consider an extended version of the unmanned aerial vehicle (UAV) motion planning problem from Badings et al. (2022a), which uses a dynamical model based on a system of 3 double integrators. We extend this model into a hybrid system, by considering two discrete modes relating to different levels of noise (a low wind speed and high wind speed, respectively), and model the system under Assumption A, with known probabilities of mode switches.

The specification requires reaching a goal set (highlighted in green in Fig. 2) whilst avoiding some obstacles (highlighted in red). For this scenario, we choose a finite time horizon of 64 time steps. While our theoretical contributions hold for any probability distribution for the additive noise in the continuous dynamics, in this particular experiment, we sample from a Gaussian distribution.

We present the optimal paths generated with the synthesised policies in Fig. 2: the blue and orange paths denote the flight under low and high wind conditions, respectively. Paths are shown for consistent conditions (i.e. where the wind speed remains low or high). In practice, we would expect mode changes to occasionally happen mid-flight.

Scalability. The number of iMDP states equals the number of regions of the partitions for the continuous state variables, multiplied by the number of discrete modes, here resulting in 51,030 states. The number of transitions depends on the number of samples W: with 100 samples we generate an iMDP with 92.7 million transitions; with 200 samples we have 154 million transitions. Computing the set of iMDP actions is independent of the sampling and takes approximately 8.5 min, computing transition probabilities then takes 70 min, finally model checking takes 40 s.

Variable noise affects decisions. With our techniques for MJLSs, we can synthesise a controller that accounts for different noise levels at runtime, and reason about the probability of the noise level changing (so that we suddenly find ourselves in worse conditions than expected). Without the ability to reason about mode transitions, one might instead plan considering only the worst case noise. In this case, the paths chosen would be overly conservative (i.e. they would travel much further from obstacles than necessary, at the cost of longer paths and flight times). Alternatively, we might work under Assumption B, synthesising a controller that is robust to any noise distribution in a set.

4.2. Temperature Regulation in a Building

We consider again the 2-room building temperature control problem (Abate et al., 2008b) introduced in Example 1. We wish to optimise the probability of satisfying the formula $\psi = (\neg T_C)\mathsf{U}^{\leq K}(T_G)$, with goal temperature T_G between 22 and 23°C, and critical temperatures T_C being those less than 20°C or greater than 25°C. We partition the state space into 1600 regions, and use a time horizon $K = 32$. We look into two setups, one fulfilling Assumption A and the other Assumption B.

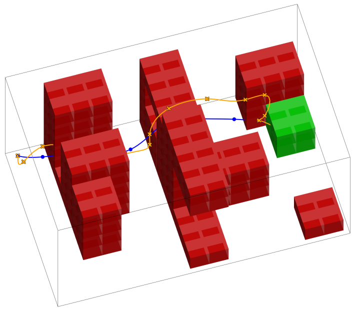

Figure 2: Optimal paths for drone dynamics, with a known probability of changing wind conditions.

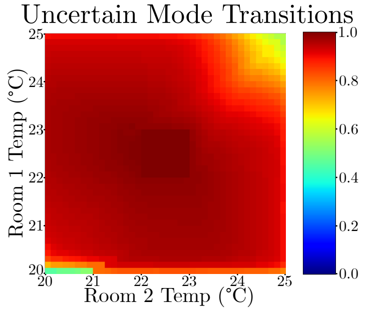

Figure 3: Lower bound on sat probabilities with uncertain transitions (Assumption A), starting in mode 1.

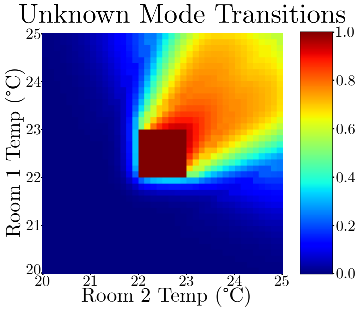

Figure 4: Lower bound probabilities of satisfaction with an unknown mode transition function (Assumption B).

Conservativism and scalability. When we wish to be robust to all possible modes, our generated iMDP is much smaller (since we now have a single iMDP, compared to one for each mode). The generated iMDP has approximately 18 times fewer transitions than the iMDP generated for Fig. 3, and PRISM takes only 1.1s to carry out model checking. However, as seen in Fig. 4, our lower bounds are much more conservative, with probabilities greater than 0.5 only encountered when both rooms start at temperatures above the goal region.

More general PCTL formulae. We have argued that our technique can handle general PCTL specifications. In Appendix C we provide an example using the rich PCTL formula

$$\mathsf{P}_{\geq 0.6}[\mathsf{X}\mathsf{P}_{\leq 0.5}(\neg T_C \mathsf{U}^{\leq K-1}(T_L \vee T_C))] \wedge \mathsf{P}_{\geq 0.9}[\neg T_C \mathsf{U}^{\leq K} T_G]. \tag{8}$$

This formula requires states that (1) can be heated to a goal temperature T_G, avoiding the critical temperatures T_C; and (2) can reach a state in the next time step, that is able to avoid reaching an unwanted temperature T_L or critical temperature T_C. This setup allows us to find states that satisfy different goals according to the requirements at specific times, with different policies corresponding to different goals. At runtime, the policy associated with the relevant goal can be used.

5. Conclusions

We have presented a method for synthesizing certifiably correct controllers for Markov jump linear systems with stochastic dynamics. We have considered both the case where an estimate of the switching probabilities is known, and that 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 Lunze and Lamnabhi-Lagarrigue (2009); Abate et al. (2010)), estimating mode-switching probabilities with the scenario approach, and dealing with a setting where matrices are only known to belong to a convex polytope, such as is done in Badings et al. (2022b) for non-hybrid systems.

References

Alessandro Abate, Alessandro D'Innocenzo, Maria Domenica Di Benedetto, and Shankar Sastry. Markov set-chains as abstractions of stochastic hybrid systems. In Magnus Egerstedt and Bud Mishra, editors, *Hybrid Systems: Computation and Control, 11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008. Proceedings*, volume 4981 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2008a. doi: 10.1007/978-3-540-78929-1_1. URL https://doi.org/10.1007/978-3-540-78929-1_1.

Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Autom.*, 44(11):2724–2734, 2008b. doi: 10.1016/j.automatica.2008.03.027. URL https://doi.org/10.1016/j.automatica.2008.03.027.

Alessandro Abate, Joost-Pieter Katoen, John Lygeros, and Maria Prandini. Approximate model checking of stochastic hybrid systems. *Eur. J. Control*, 16(6):624–641, 2010. doi: 10.3166/ejc.16.624-641. URL https://doi.org/10.3166/ejc.16.624-641.

Thom S. Badings, Alessandro Abate, Nils Jansen, David Parker, Hasan A. Poonawala, and Mariëlle Stoelinga. Sampling-based robust control of autonomous systems with non-gaussian noise. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9669–9678. AAAI Press, 2022a. URL https://ojs.aaai.org/index.php/AAAI/article/view/21201.

Thom S. Badings, Licio Romao, Alessandro Abate, and Nils Jansen. Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. *CoRR*, abs/2210.05989, 2022b. doi: 10.48550/arXiv.2210.05989. URL https://doi.org/10.48550/arXiv.2210.05989.

Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008. ISBN 978-0-262-02649-9.

Mohammed Benbrahim, Mohammed Kabbaj, and Khalid Benjelloun. Robust control under constraints of linear systems with markovian jumps. *International Journal of Control, Automation and Systems*, 14, 10 2016. doi: 10.1007/s12555-015-0121-0. URL https://doi.org/10.1007/s12555-015-0121-0.

Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Trans. Robotics*, 26(3):502–517, 2010. doi: 10.1109/TRO.2010.2044948. URL https://doi.org/10.1109/TRO.2010.2044948.

El Kébir Boukas and Hailiang Yang. Stability of discrete-time linear systems with markovian jumping parameters. *Math. Control. Signals Syst.*, 8(4):390–402, 1995. doi: 10.1007/BF01209692. URL https://doi.org/10.1007/BF01209692.

Hongbin Cai, Ping Li, Chengli Su, and Jiangtao Cao. Robust model predictive control for a class of discrete-time markovian jump linear systems with operation mode disordering. *IEEE Access*, 7: 10415–10427, 2019. doi: 10.1109/ACCESS.2019.2891506. URL https://doi.org/10.1109/ACCESS.2019.2891506.

Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annu. Rev. Control.*, 33(2):149–157, 2009. doi: 10.1016/j.arcontrol.2009.07.001. URL https://doi.org/10.1016/j.arcontrol.2009.07.001.

Nathalie Cauchi, Luca Laurenti, Morteza Lahijanian, Alessandro Abate, Marta Kwiatkowska, and Luca Cardelli. Efficiency through uncertainty: scalable formal synthesis for stochastic hybrid systems. In Necmiye Ozay and Pavithra Prabhakar, editors, *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2019, Montreal, QC, Canada, April 16-18, 2019*, pages 240–251. ACM, 2019. doi: 10.1145/3302504.3311805. URL https://doi.org/10.1145/3302504.3311805.

Edmund M. Clarke, Ansgar Fehnker, Zhi Han, Bruce H. Krogh, Olaf Stursberg, and Michael Theobald. Verification of hybrid systems based on counterexample-guided abstraction refinement. In Hubert Garavel and John Hatcliff, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 9th International Conference, TACAS 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, volume 2619 of *Lecture Notes in Computer Science*, pages 192–207. Springer, 2003. doi: 10.1007/3-540-36577-X_14. URL https://doi.org/10.1007/3-540-36577-X_14.

Rafael F. Cunha, Gabriela W. Gabriel, and José Claudio Geromel. Robust partial sampled-data state feedback control of markov jump linear systems. *Int. J. Syst. Sci.*, 50(11):2142–2152, 2019. doi: 10.1080/00207721.2019.1647305. URL https://doi.org/10.1080/00207721.2019.1647305.

Daniela Pucci de Farias, José Claudio Geromel, João B. R. do Val, and Oswaldo L. V. Costa. Output feedback control of markov jump linear systems in continuous-time. *IEEE Trans. Autom. Control.*, 45(5):944–949, 2000. doi: 10.1109/9.855557. URL https://doi.org/10.1109/9.855557.

Oswaldo Luiz Valle do Costa, Ricardo Paulino Marques, and Marcelo Dutra Fragoso. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer, London, 2005. ISBN 978-1-85233-761-2 978-1-84628-082-5. doi: 10.1007/b138575. URL https://doi.org/10.1007/b138575.

Oswaldo Luiz do Valle Costa and Marcelo D. Fragoso. Discrete-time lq-optimal control problems for infinite markov jump parameter systems. *IEEE Trans. Autom. Control.*, 40(12):2076–2088, 1995. doi: 10.1109/9.478328. URL https://doi.org/10.1109/9.478328.

Oswaldo Luiz do Valle Costa, Marcelo D. Fragoso, and Marcos Garcia Todorov. 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. doi: 10.1109/TAC.2014.2366253. URL https://doi.org/10.1109/TAC.2014.2366253.

Gabriela W. Gabriel and José Claudio Geromel. Performance evaluation of sampled-data control of markov jump linear systems. *Autom.*, 86:212–215, 2017. doi: 10.1016/j.automatica.2017.08.015. URL https://doi.org/10.1016/j.automatica.2017.08.015.

Ernst Moritz Hahn, Tingting Han, and Lijun Zhang. Synthesis for PCTL in parametric markov decision processes. In Mihaela Gheorghiu Bobaru, Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors, *NASA Formal Methods - Third International Symposium, NFM 2011, Pasadena, CA, USA, April 18-20, 2011. Proceedings*, volume 6617 of *Lecture Notes in Computer Science*, pages 146–161. Springer, 2011. doi: 10.1007/978-3-642-20398-5_12. URL https://doi.org/10.1007/978-3-642-20398-5_12.

Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects Comput.*, 6(5):512–535, 1994. doi: 10.1007/BF01211866. URL https://doi.org/10.1007/BF01211866.

João Pedro Hespanha, Payam Naghshtabrizi, and YongGang Xu. A survey of recent results in networked control systems. *Proc. IEEE*, 95(1):138–162, 2007. doi: 10.1109/JPROC.2006.887288. URL https://doi.org/10.1109/JPROC.2006.887288.

Li-Sheng Hu, Peng Shi, and Paul Martin Frank. Robust sampled-data control for markovian jump linear systems. *Autom.*, 42(11):2025–2030, 2006. doi: 10.1016/j.automatica.2006.05.029. URL https://doi.org/10.1016/j.automatica.2006.05.029.

Baoping Jiang, Zhengtian Wu, and Hamid Reza Karimi. A traverse algorithm approach to stochastic stability analysis of markovian jump systems with unknown and uncertain transition rates. *Applied Mathematics and Computation*, 422:126968, 2022. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2022.126968. URL https://www.sciencedirect.com/science/article/pii/S0096300322000546.

John C. Knight. Safety critical systems: challenges and directions. In Will Tracz, Michal Young, and Jeff Magee, editors, *Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, 19-25 May 2002, Orlando, Florida, USA*, pages 547–550. ACM, 2002. doi: 10.1145/581339.581406. URL https://doi.org/10.1145/581339.581406.

Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. doi: 10.1007/978-3-642-22110-1_47. URL https://doi.org/10.1007/978-3-642-22110-1_47.

Abolfazl Lavaei, Sadegh Soudjani, Alessandro Abate, and Majid Zamani. Automated verification and synthesis of stochastic hybrid systems: A survey. *CoRR*, abs/2101.07491, 2021. URL https://arxiv.org/abs/2101.07491.

Wenbai Li, Yu Xu, and Huaizhong Li. 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. doi: 10.1049/iet-spr.2012.0325. URL https://doi.org/10.1049/iet-spr.2012.0325.

Lars Lindemann, Haimin Hu, Alexander Robey, Hanwen Zhang, Dimos V. Dimarogonas, Stephen Tu, and Nikolai Matni. Learning hybrid control barrier functions from data. In Jens Kober, Fabio Ramos, and Claire J. Tomlin, editors, *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 1351–1370. PMLR, 2020. URL https://proceedings.mlr.press/v155/lindemann21a.html.

Yuriy Zacchia Lun, Jack Wheatley, Alessandro D'Innocenzo, and Alessandro Abate. Approximate abstractions of markov chains with interval decision processes. In Alessandro Abate, Antoine Girard, and Maurice Heemels, editors, *6th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2018, Oxford, UK, July 11-13, 2018*, volume 51 of *IFAC-PapersOnLine*, pages 91–96. Elsevier, 2018. doi: 10.1016/j.ifacol.2018.08.016. URL https://doi.org/10.1016/j.ifacol.2018.08.016.

Jan Lunze and Françoise Lamnabhi-Lagarrigue, editors. *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, Cambridge, 2009. ISBN 978-0-521-76505-3. doi: 10.1017/CBO9780511807930. URL https://doi.org/10.1017/CBO9780511807930.

Eugenio Moggi, Amin Farjudian, Adam Duracz, and Walid Taha. Safe & robust reachability analysis of hybrid systems. *Theor. Comput. Sci.*, 747:75–99, 2018. doi: 10.1016/j.tcs.2018.06.020. URL https://doi.org/10.1016/j.tcs.2018.06.020.

Cecília F. Morais, Jonathan M. Palma, Pedro L. D. Peres, and Ricardo C. L. F. Oliveira. 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. doi: 10.1016/j.automatica.2018.06.014. URL https://doi.org/10.1016/j.automatica.2018.06.014.

Ameneh Nejati, Sadegh Soudjani, and Majid Zamani. Compositional construction of control barrier functions for continuous-time stochastic hybrid systems. *Autom.*, 145:110513, 2022. doi: 10.1016/j.automatica.2022.110513. URL https://doi.org/10.1016/j.automatica.2022.110513.

Sangwoo Park, Erchin Serpedin, and Khalid A. Qaraqe. Gaussian assumption: The least favorable but the most useful [lecture notes]. *IEEE Signal Process. Mag.*, 30(3):183–186, 2013. doi: 10.1109/MSP.2013.2238691. URL https://doi.org/10.1109/MSP.2013.2238691.

André Platzer. Logics of dynamical systems. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 13–24. IEEE Computer Society, 2012. doi: 10.1109/LICS.2012.13. URL https://doi.org/10.1109/LICS.2012.13.

Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. Polynomial-time verification of PCTL properties of mdps with convex uncertainties. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 527–542. Springer, 2013. doi: 10.1007/978-3-642-39799-8_35. URL https://doi.org/10.1007/978-3-642-39799-8_35.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi: 10.1002/9780470316887. URL https://doi.org/10.1002/9780470316887.

Stefan Ratschan and Zhikun She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In Manfred Morari and Lothar Thiele, editors, *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, volume 3414 of *Lecture Notes in Computer Science*, pages 573–589. Springer, 2005. doi: 10.1007/978-3-540-31954-2_37. URL https://doi.org/10.1007/978-3-540-31954-2_37.

Alexander Robey, Lars Lindemann, Stephen Tu, and Nikolai Matni. Learning robust hybrid control barrier functions for uncertain systems. In Raphaël M. Jungers, Necmiye Ozay, and Alessandro Abate, editors, *7th IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2021, Brussels, Belgium, July 7-9, 2021*, volume 54 of *IFAC-PapersOnLine*, pages 1–6. Elsevier, 2021. doi: 10.1016/j.ifacol.2021.08.465. URL https://doi.org/10.1016/j.ifacol.2021.08.465.

Paulo Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009. ISBN 978-1-4419-0223-8. URL http://www.springer.com/mathematics/applications/book/978-1-4419-0223-8.

Engang Tian, Dong Yue, and Guoliang Wei. Robust control for markovian jump systems with partially known transition probabilities and nonlinearities. *J. Frankl. Inst.*, 350(8):2069–2083, 2013. doi: 10.1016/j.jfranklin.2013.05.011. URL https://doi.org/10.1016/j.jfranklin.2013.05.011.

Lixian Zhang and El Kébir Boukas. Stability and stabilization of markovian jump linear systems with partly unknown transition probabilities. *Autom.*, 45(2):463–468, 2009. doi: 10.1016/j.automatica.2008.08.010. URL https://doi.org/10.1016/j.automatica.2008.08.010.

Appendix A. Abstractions for a single mode

States. We discretise a subset $\mathcal{X} \subset \mathbb{R}^n$ of the continuous state space through a partition $\mathcal{R} = \{R_1, \ldots, R_p\}$ into $|\mathcal{R}|$ convex polytopes. To capture any state $x \notin \mathcal{X}$, we define one additional region $\mathbb{R}^n \setminus \mathcal{R} = R_a$. We then define a function $L \colon \mathbb{R}^n \to \{1, 2, \ldots, |\mathcal{R}| + 1\}$ which maps a continuous state x to the index i of a region in the partition $\mathcal{R}$ for which $x \in R_i$, or to the absorbing region if $x \notin \mathcal{X}$ (which has index $|\mathcal{R}| + 1$). The inverse mapping is denoted by $R_i = L^{-1}(i) \subset \mathcal{X}$.

This partition into $|\mathcal{R}|$ polytopes, and an absorbing region R_a, provides a natural set of states $\mathcal{S}$ for our MDP, with the absorbing state only having a transition back to itself. To transfer the properties from the Markov jump system to the abstraction, we also make the following assumption between the labelling function on atomic propositions and the mapping $L(x)$, which requires that the atomic propositions for the PCTL do not overlap.

Assumption 1 (Discrete region alignment) *The state space discretization is such that*

$$L(x) \models p \implies L(x') \models p, \forall x, x' : L(x) = L(x').$$

Enabled Actions. We define a finite set of actions $\mathcal{A} = \{a_1, \ldots, a_q\}, q \in \mathbb{N}$ and associate each discrete action with a target point $\{d_1, \ldots, d_q\}, d_i \in \mathcal{X}$. Let us define the noiseless successor state $\hat{x}(k+1) = A_{r(k)}x(k) + B_{r(k)}u(k) + q_{r(k)}$, which coincides with Eq. (2) for $w_{r(k)} = 0$. Taking an action a_j then means that we drive $\hat{x}(k+1)$ to target point d_j, by deriving a control input $u(k)$ such that $\hat{x}(k+1) = d_i$. To obtain an abstraction that is sound, we only enable an action a_j in some state $s_i \in \mathcal{S}$ if that action is feasible from each continuous state x in the corresponding region R_i. To this end, we introduce the backward reachable set for a target point d_j in some dynamics $i \in \mathcal{N}$.

$$\mathcal{G}(d_j) = \{x \in \mathbb{R}^n \mid d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, i \in \mathcal{N}\}, \tag{9}$$

which is set of all points which can reach d_j given the current dynamics. We then only enable action d_j in states s_{d_j} which are *fully contained* in the backward reachable set:

$$s_{d_j} = \{s_i \in \mathcal{S} \colon x \in \mathcal{G}(d_j) \, \forall x \in R_i\}. \tag{10}$$

The next assumption asserts that the regions in $\mathcal{R}$ can be contained in the backward reachable set.

Assumption 2 (Non-empty backward reachable sets) *The backward reachable set $\mathcal{G}(d_j)$ has a non-empty interior, which implies that matrices $B_i, i \in \mathcal{N}$ have full row rank, i.e. rank$(B_i) = n, \forall i \in \mathcal{N}$, where $n = dim(\mathbf{x})$.*

Remark 5 *The above assumption is somewhat restrictive on the class of models we can apply our method to, but this can be relaxed using techniques recently developed in Badings et al. (2022b).*

Transition Probabilities. The transition probability intervals for individual modes are found using a sampling-based approach, based on a finite set of W observations of the noise $w^{(i)}_{r(k)}(k), i = 1, \ldots, W$. The process noise $w_{r(k)}(k) \in \Delta_{r(k)}$ is defined on a probability space $(\Delta_{r(k)}, \mathcal{D}_{r(k)}, \mathbb{P}_{r(k)})$, with σ-algebra $\mathcal{D}_{r(k)}$ and probability measure $\mathbb{P}_{r(k)}$ defined over $\mathcal{D}_{r(k)}$. Each sample has a unique index $i = 1, \ldots, W$ and is associated with a possible successor state $x(k+1) = \hat{x}(k+1) + w^{(i)}_{r(k)}(k)$. We assume that we are able to obtain these samples at low cost from experimental data or simulations.

Assumption 3 *The noise samples $w^{(i)}_{r(k)}(k), i = 1, \dots, W$ are i.i.d. elements from $(\Delta_{r(k)}, \mathbb{P}_{r(k)})$, and are independent of time.*

Since the noise samples are i.i.d. the set of W samples is a random element from the probability space $\Delta^W_{r(k)}$, equipped with the product probability $\mathbb{P}^W_{r(k)}$.

We use methods from the scenario approach (Campi et al., 2009) to compute *probability intervals* which contain the true transition probability with high confidence. To this end, we compute upper and lower bounds for every probability $P(s_i, a_l)(s_j)$ and then use these to define the transition probability intervals for the iMDP. As the intervals are PAC, this iMDP will be a robust abstraction of the individual mode dynamics.

First, we introduce the concept of *violation probability*, which measures the probability that a generated successor state is *not* in a given region.

Definition 6 *The violation probability $P_{w_{r(k)}}(x(k+1) \notin R_j)$ is the probability that a successor state $x(k+1)$ is not in region R_j, and is given by*

$$\begin{aligned} P_{w_{r(k)}(k)}(x(k+1) \notin R_j) &= \mathbb{P}\{w_{r(k)}(k) \in \Delta_{r(k)} : \hat{x}(k) + w_{r(k)}(k) \notin R_j\} \\ &= 1 - P_{w_{r(k)}(k)}(x(k+1) \in R_j). \end{aligned} \tag{11}$$

It is important to note that, in our case $P(s_i, a_l)(s_j) = P_{w_{r(k)}(k)}(x(k+1) \in R_j)$.

The scenario approach allows us to bound the risk that the optimal point of a so-called *scenario optimization problem* does not belong to a feasible set $\tilde{R}$, defined by a set of (possibly infinite) constraints, when we are only able to sample a subset of those constraints.

By formulating this optimization problem such that $\tilde{R}$ is closely related to a region R_j, we can obtain upper and lower bounds on the violation probability. This means that we can adapt results from the theory for the scenario approach to compute transition probability intervals for our abstractions.

Thus, given a transition (s_i, a_l, s_j), and a corresponding number of samples W^{out}_j which lie outside region R_j defined as

$$W^{\text{out}}_j = |\{i \in \{1, \dots, W\} \mid (\hat{x}(k+1) + w^{(i)}_{r(k)}(k)) \in R_j\}|. \tag{12}$$

We can calculate an interval that contains $P(s_i, a_l)(s_j)$ with at least some pre-defined confidence level.

Theorem 7 (PAC probability intervals (Theorem 1 in Badings et al. (2022a))) *For $W \in \mathbb{N}$ samples of the noise, fix a confidence parameter $\beta \in (0, 1)$. Given W^{out}_j, the transition probability $P(s_i, a_l)(s_j)$ is bounded by*

$$\mathbb{P}^W\{\underline{p} \leq P(s_i, a_l)(s_j) \leq \overline{p}\} \geq 1 - \beta, \tag{13}$$

where $\underline{p} = 0$ if $W^{out}_j = W$, and otherwise $\underline{p}$ is the solution of

$$\frac{\beta}{2W} = \sum_{i=0}^{W^{out}_j} \binom{W}{i} (1 - \underline{p})^i \underline{p}^{W-i}, \tag{14}$$

and $\overline{p} = 1$ *if* $W_j^{out} = 0$*, and otherwise* $\overline{p}$ *is the solution of*

$$\frac{\beta}{2W} = 1 - \sum_{i=0}^{W_j^{out}-1} \binom{W}{i} (1-\overline{p})^i \overline{p}^{W-i}. \tag{15}$$

This claim holds for any Δ and $\mathbb{P}$, meaning that we can bound the transition probabilities even for an unknown probability distribution.

This then provides the transition probability intervals within a mode. These intervals then need to be combined between the modes to define the complete transition probabilities for the continuous state and (possibly) the discrete mode of the system.

Appendix B. Proofs

B.1. Proof of Theorem 3

Consider a state x in the intersection of backward reachable sets:

$$x \in \bigcap_{i \in \mathcal{N}} \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\}.$$

Since this state is in the intersection of sets, then, by definition, it must also be contained in each set individually:

$$x \in \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\}, \forall i \in \mathcal{N}.$$

Since this holds for all modes $i \in \mathcal{N}$, and it suffices that there is a (possibly different) control input $u \in \mathcal{U}$ that can reach d_j in each mode, then we can say that

$$x \in \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, \forall i \in \mathcal{N}\}.$$

If we consider a state x' not contained in the intersection of backward reachable sets:

$$x' \notin \bigcap_{i \in \mathcal{N}} \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\},$$

then, by the definition of set intersections, there must be at least one mode such that

$$\nexists u \in \mathcal{U} : d_j = A_i x' + B_i u + q_i.$$

Then x' cannot be contained in the set

$$\{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, \forall i \in \mathcal{N}\},$$

since there is at least one mode for which $d_j = A_i x + B_i u + q_i, u \in \mathcal{U}$, does not hold.

Thus, we can conclude that

$$\bigcap_{i \in \mathcal{N}} \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}\} = \{x \in \mathbb{R} | d_j = A_i x + B_i u + q_i, u \in \mathcal{U}, \forall i \in \mathcal{N}\},$$

completing the proof.

B.2. Proof of Theorem 4

In our construction of our iMDP we have used Theorem 3, to ensure, we only enable actions that can be reached in any possible mode of the system. Thus, when we do uncover our true mode, we can be certain that our chosen action will be associated with a valid control input.

Then, we have constructed an interval $[\underline{P}, \overline{P}]$ which contains the transition intervals of every individual iMDP $[\underline{P_i}, \overline{P_i}]$:

$$\underline{P_i} \in [\underline{P}, \overline{P}], \overline{P_i} \in [\underline{P}, \overline{P}].$$

When we uncover the true mode i we are in, the true probability P associated with that mode, will lie in the interval $[\underline{P_i}, \overline{P_i}]$ with confidence $1-\beta$ (by Theorem 7). Since we have confidence $1-\beta$ that $P \in [\underline{P_i}, \overline{P_i}]$, and $\forall P \in [\underline{P_i}, \overline{P_i}], P \in [\underline{P}, \overline{P}]$ then with confidence at least $1-\beta$, $P \in [\underline{P}, \overline{P}]$.

Therefore, our abstraction under Assumption B, will have actions correctly enabled, and transition probabilities which will lie in the constructed interval with a given confidence. Thus, we can conclude that our abstraction is sound.

B.3. Details on Def. 2

States and actions are soundly generated as the Cartesian product of states and actions in the individual iMDPs. Then transition intervals are constructed as

$$\mathcal{P}_\times((r,s)(l,a))((r',s')) = [\underline{P}(r,l)(r') \cdot \underline{P_r}(s,a)(s'),\ \overline{P}(r,l)(r') \cdot \overline{P_r}(s,a)(s')].$$

Consider the true probabilities $P(r,l)(r')$ and $P_r(s,a)(s') = \mathbb{P}(s,a,s'|r,l,r') = \mathbb{P}(s,a,s' \mid r)$ (since the state transition is independent of the discrete mode action and next mode. Then, in a state (s,r) the probability of transition to state (s',r') is $P(r,l)(r') \cdot P_r(s,a)(s')$. Since $P(r,l)(r') \in [\underline{P}(r,l)(r'), \overline{P}(s,a)(s')]$ and $P_r(s,a)(s') \in [\underline{P_r}(s,a)(s'), \overline{P_r}(s,a)(s')]$ then, by the properties of the real numbers, $P(r,l)(r) \cdot P_r(s,a)(s) \in [\underline{P}(r,l)(r') \cdot \underline{P_r}(s,a)(s),\ \overline{P}(r,l)(r') \cdot \overline{P_r}(s,a)(s')]$. Since $a < b, 0 < c \implies ac < bc$.

Appendix C. Experiment Details and Additional Results

C.1. Experiment Details

C.1.1. UAV MOTION PLANNING

For the UAV motion planning problem, we considered a 6 dimensional state vector defined as $x = (p_x, p_y, p_z, v_x, v_y, v_z)^\top \in \mathbb{R}^6$, with p_i and v_i denoting the position and velocity in direction i. The dynamics follow a double integrator model, so that the resulting state equations are:

$$x(k+1) = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^2}{2} \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} u(k) + w(k), \tag{16}$$

where $w(k)$ is the noise term arising from turbulence, and $u(k) \in \mathbb{R}^3$ is the acceleration, and control input, which is constrained to the interval $[-4, 4]$.

Since the model in Eq. (16) is not fully actuated (it has only 3 controls, with a state space of 6) we group every two time steps together to rewrite the model as

$$x(k+2) = \bar{A}x(k) + \bar{B}\begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} Aw(k) \\ w(k+1) \end{bmatrix}. \tag{17}$$

With $\bar{A} = A^2$ and $\bar{B} = \begin{bmatrix} AB & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}$. Then, the two control inputs are placed into a single vector, which now has dimension 6.

In this setup, the only difference between the two modes is the distribution of the noise, in the low wind speed mode, this is distributed according to a zero-mean Gaussian with standard deviation of 0.15. In the high wind speed mode, the standard deviation is instead 1.5. The jumps between these modes are modelled with a precisely known MDP, with a 10% chance of switching from low to high wind speed, and then a 30% chance of switching from high to low.

The objective, is to reach the goal region (highlighted in green), whilst avoiding critical regions (highlighted in red), within 64 time steps (or 32 time steps with the amended model).

C.1.2. TEMPERATURE REGULATION IN A BUILDING

The equations for the temperature regulation are defined using the following matrices:

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

With a state $x = [T_1, T_2]^\top \in \mathbb{R}^2$ referring to the temperature in each room, and a control input $u \in \mathbb{R}^2$ with each element constrained to the interval $[0, 1]$ referring to the amount of power supplied to the heater in room 1 and 2 respectively.

The exact constants used in our experiments are $a_{12} = 0.022, b_1 = b_2 = 0.0167, k_f = 0.8, k_r = 0.4, x_a = 6$. The noise was distributed according to a zero-mean Gaussian with standard deviation 0.2.

We investigate 3 problems with this model.

In the first two, we are interested in reaching a goal region defined as a temperature of between 22 and 23°C in both rooms, whilst avoiding critical temperatures below 20°C, or above 25°C. In the first problem, we model our system under Assumption A, with a known interval probability of mode switching failing defined to be 0.1-20%. In the second, we model our system under Assumption B, and attempt to be robust to any mode. Finally, we investigate a richer PCTL formula outlined in Eq. (8), in this case, the goal and critical temperatures remain the same, and an additional region T_L is defined for temperatures below 21°C in room 1.

C.2. Additional Experimental Outcomes

We present here a few additional experimental results. First, we present an experiment similar to that in Fig. 3, but instead with mode transitions happening at random. This change results in slightly lower satisfaction guarantees, since we have cannot choose the next mode.

We then present the results for states in mode 2, corresponding to the results in Fig. 3, which only showed the results for an initial mode 1.

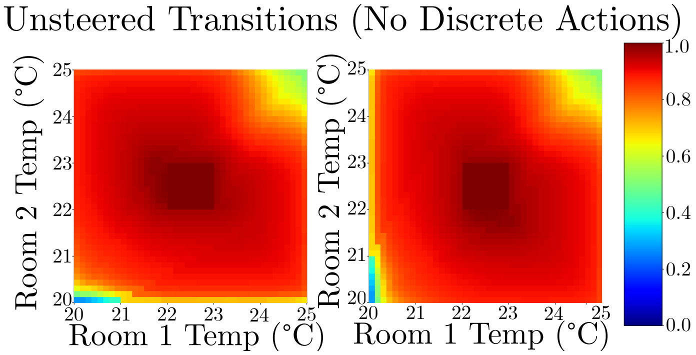

Figure 5: Results for a temperature control problem, with mode transitions that are driven by an MC, with probabilities known to be in the interval 40-60%.

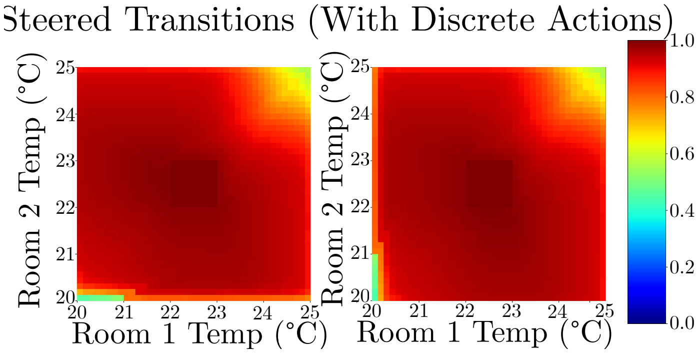

Figure 6: Lower bound probabilities on reaching goal temperature, as in Fig. 3, but presented for both modes.

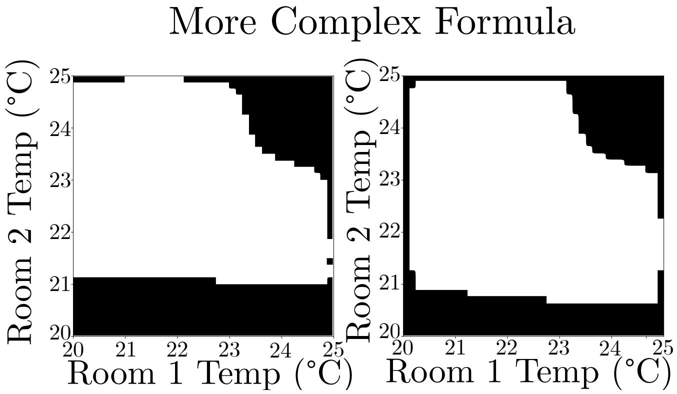

Figure 7: Satisfying states of Eq. (8), in both modes, states presented in white are those which satisfy the formula.

Finally, we provide an additional figure which demonstrates our ability to satisfy the richer formula provided in Eq. (8).