Robust Optimization for Hybrid MDPs with State-dependent Noise

Zahra Zamani
ANU & NICTA
Canberra, Australia
zahra.zamani@anu.edu.au

Scott Sanner
NICTA & ANU
Canberra, Australia
ssanner@nicta.com.au

Karina Valdivia Delgado
University of Sao Paulo
Sao Paulo, Brazil
kvd@ime.usp.br

Leliane Nunes de Barros
University of Sao Paulo
Sao Paulo, Brazil
leliane@ime.usp.br

Abstract

Recent advances in solutions to hybrid MDPs with discrete and continuous state and action spaces have significantly extended the class of MDPs for which exact solutions can be derived, albeit at the expense of a restricted transition noise model. In this paper, we work around limitations of previous solutions by adopting a robust optimization approach in which Nature is allowed to adversarially determine transition noise within automatically-derived confidence intervals. This allows one to derive an optimal policy with an arbitrary (user-specified) level of success probability and significantly extends the class of transition noise models for which hybrid MDPs can be solved. This work also significantly extends results for the related "chance-constrained" approach in stochastic hybrid control to accommodate state-dependent noise. We demonstrate our approach working on a variety of hybrid MDPs taken from AI planning, operations research, and control theory, noting that this is the first time optimal robust solutions have been automatically derived for such problems.

1 Introduction

Many real-world sequential decision-making problems are naturally modeled with both discrete and continuous (hybrid) state and action spaces. When state transitions are stochastic, these problems can be modeled as Hybrid Markov Decision Processes (HMDPs), which have been studied extensively in AI planning [Boyan and Littman, 2001; Feng *et al.*, 2004; Li and Littman, 2005; Kveton *et al.*, 2006; Marecki *et al.*, 2007; Meuleau *et al.*, 2009; **?**] as well as control theory [Henzinger *et al.*, 1997; Hu *et al.*, 2000; De Schutter *et al.*, 2009] and operations research [Puterman, 1994]. However, all previous solutions to hybrid MDPs either take an approximation approach or restrict stochastic noise on continuous transitions to be state-independent or discretized (i.e., requiring continuous transitions to be a finite mixture over deterministic transitions).

Unfortunately, each of these assumptions can be quite limiting in practice when strong *a priori* guarantees on performance are required in the presence of general forms of state-dependent noise. For example, in a UAV NAVIGATION problem [], a human controller must be aware of all positions from which a UAV with a given amount of fuel reserves can return to its landing strip with high probability of success given known areas of (state-dependent) turbulence and weather events. In a SPACE TELESCOPE CONTROL problem [], one must carefully manage inertial moments and rotational velocities as the telescope maneuvers between different angular orientations and zoom positions, where noise margins increase when the telescope is in unstable positions (extended zooms). And lastly, in a RESERVOIR CONTROL problem, one must manage reservoir levels to ensure a sufficient water supply for a population while avoiding overflow conditions subject to uncertainty over daily rainfall amounts. In all of these problems, there is no room for error: a UAV crash, a space telescope spinning uncontrollably, or a flooded reservoir can all cause substantial physical, monetary, and/or environmental damage. What is needed are robust solutions to these problems that are cost-optimal while guaranteed not to exceed a prespecified margin of error.

To achieve cost-optimal robust solutions we build on ideas used in the chance-constrained control literature [Schwarm and Nikolaou, 1999; Li *et al.*, 2002; Ono and Williams, 2008; Blackmore *et al.*, 2011] that maintain confidence intervals on (multivariate) noise distributions and ensure that all reachable states are within these noise margins. However, all previous methods restrict either to linear systems with Gaussian uncertainty and state-independent noise or otherwise resort to approximation techniques. Furthermore, as these works are all inherently focused on control from a given initial state, they are unable to prove properties such as *robust controllability*, i.e., what states have a policy that can achieve a given cost with high certainty over some horizon?

In this work, we adopt a robust optimization receding horizon control approach in which Nature is allowed to adversarially determine transition noise within automatically-derived confidence intervals. This permits us to find *exact* solutions for a wide range of HMDPs that previously as well as allowing us to answer questions of *robust controllability* in very general state-dependent continuous noise settings. Altogether, this work significantly extends previous results in both the HMDP literature in AI and robust hybrid control literature and permits the solution of a new class of robust HMDP control problems.

2 Robust Continuous State and Action MDPs

We first formally introduce the framework of Robust Continuous State and Action Markov decision processes (RCSA-MDPs) extended from CSA-MDPs [?]. The optimal solution is then defined by a Robust Dynamic Programming (RDP) approach.

2.1 Factored Representation

A RCSA-MDP is modelled using state variables $(\vec{b}, \vec{x}) = (b_1, \ldots, b_a, x_1, \ldots, x_c)$ where each $b_i \in \{0, 1\}$ $(1 \le i \le a)$ represents discrete boolean variables and each $x_j \in \mathbb{R}$ $(1 \le j \le c)$ is continuous. To model uncertainty in RCSA-MDPs we assume intermediate noise variables $\vec{n} = n_1, \ldots, n_e$ which can be state dependent. Both discrete and continuous actions are represented by the set $A = \{a_1(\vec{y}_1), \ldots, a_p(\vec{y}_d)\}$, where $\vec{y}_k \in \mathbb{R}^{|\vec{y}_k|}$ $(1 \le k \le d)$ denote continuous parameters for action a_k.

Given a state $(\vec{b}, \vec{x})$ and an executed action $a(\vec{y})$ at this state, a reward function $R(\vec{b}, \vec{x}, a, \vec{y})$ specifies the immediate reward at this state. The probability of the next state $(\vec{b}', \vec{x}')$ is defined by a joint state transition model $P(\vec{b}', \vec{x}'|\vec{b}, \vec{x}, a, \vec{y}, \vec{n})$ which depends on the current state, action and noise variables. We assume stochasticity in RCSA-MDPs using a state dependent noise function $N(\vec{n}, \vec{b}, \vec{x})$ where each noise variable (i.e. stochastic) $\vec{n}$ is bounded by some convex region on state variables $(\vec{b}, \vec{x})$. Non-deterministic constraint functions define the possible values for each n_k where we assume $+\infty$ for all illegal values of noise and $-\infty$ for legal values. We show later how this general representation allows RSCA-MDPs to naturally determine the minimum noise in any problem.

A policy $\pi(\vec{b}, \vec{x})$ at this state specifies the action $a(\vec{y}) = \pi(\vec{b}, \vec{x})$ to take at this state. An optimal sequence of finite horizon policies $\Pi^* = (\pi^{*,1}, \ldots, \pi^{*,H})$ is desired such that given the initial state $(\vec{b}_0, \vec{x}_0)$ at $h = 0$ and a discount factor $\gamma,\ 0 \le \gamma \le 1$, the expected sum of discounted rewards over horizon $h \in H; H \ge 0$ is maximized:

$$V^{\Pi^*}(\vec{b}, \vec{x}) = E_{\Pi^*}\left[\sum_{h=0}^{H} \gamma^h \cdot r^h \Big| \vec{b}_0, \vec{x}_0\right]. \qquad (1)$$

where r^h is the reward obtained at horizon h following the optimal policy.

Similar to the dynamic Bayes net (DBN) structure of CSA-MDPs [?] we assume *synchronic arcs* (variables that condition on each other in the same time slice) from $\vec{b}$ to $\vec{x}$ and from the noise variables $\vec{n}$ to both $\vec{b}$ and $\vec{x}$ but not within the binary $\vec{b}$ or continuous variables $\vec{x}$.Thus the factorized joint transition model is defined as

$$P(\vec{b}', \vec{x}'|\vec{b}, \vec{x}, a, \vec{y}, \vec{\epsilon}) = \prod_{i=1}^{a} P(b'_i|\vec{b}, \vec{x}, a, \vec{y}, , \vec{n}) \prod_{j=1}^{c} P(x'_j|\vec{b}, \vec{b}', \vec{x}, a, \vec{y}, \vec{n}) \qquad (2)$$

For binary variables b_i $(1 \le i \le a)$ the conditional probability $P(b'_i|\vec{b}, \vec{x}, a, \vec{y}, \vec{n})$ is defined as conditional probability functions (CPFs) which are not tabular. For continuous variables x_j $(1 \le j \le c)$, the CPFs $P(x'_j|\vec{b}, \vec{b}', \vec{x}, a, \vec{y}, \vec{n})$ are represented with *piecewise linear equations* (PLEs) that condition on the action, noise, current state, and previous state variables with piecewise conditions that may be arbitrary logical combinations of $\vec{b}$, $\vec{b}'$ and linear inequalities over $\vec{x}$.

We allow the reward function $R(\vec{b}, \vec{x}, a, \vec{y})$ to be a general piecewise linear function (boolean or linear conditions and linear values) or a piecewise quadratic function of univariate state and a linear function of univariate action parameters. These constraints ensure piecewise linear boundaries that can be checked for consistency using a linear constraint feasibility checker, which we will later see is crucial for efficiency.

2.2 Robust Dynamic Programming

Given the general dynamic programming approach for obtaining the optimal policy, we extend the value iteration algorithm [Bellman, 1957] to a Robust dynamic programming algorithm for continuous state and actions. Initializing $V^0(\vec{b}, \vec{x}) = 0$) the algorithm builds the h-stage-to-go value functions $V^h(\vec{b}, \vec{x})$.

The quality $Q_a^h(\vec{b}, \vec{x}, \vec{y}, \vec{n})$ of taking action $a(\vec{y})$ in state $(\vec{b}, \vec{x})$ and acting so as to obtain $V^{h-1}(\vec{b}, \vec{x})$ is defined as the following:

$$Q_a^h(\vec{b}, \vec{x}, \vec{y}, \vec{n}) = \max_{n_1, \ldots, n_k} N(\vec{n}, \vec{b}, \vec{x}) \cdot \Bigg[R(\vec{b}, \vec{x}, a, \vec{y}) + \gamma \cdot \sum_{\vec{b}'} \int \left(\prod_{i=1}^{a} P(b'_i|\vec{b}, \vec{x}, a, \vec{y}, \vec{n}) \prod_{j=1}^{c} P(x'_j|\vec{b}, \vec{b}', \vec{x}, a, \vec{y}, \vec{n}) \right) V^{h-1}(\vec{b}', \vec{x}') d\vec{x}' \Bigg] \qquad (3)$$

Here the noise function $N(\vec{n}, \vec{b}, \vec{x})$ is is used to in-cooperate the noise variable $n_1, \ldots, n_k$ maximally in the original Q-value definition of value iteration. Given $Q_a^h(\vec{b}, \vec{x}, \vec{y}, \vec{n})$ for each $a \in A$, we can proceed to define the h-stage-to-go value function so as to minimize the noise as follows:

$$V^h(\vec{b}, \vec{x}) = \max_{a \in A} \max_{\vec{y} \in \mathbb{R}^{|\vec{y}|}} \min_{\vec{n}} \left\{ Q_a^h(\vec{b}, \vec{x}, \vec{y}, \vec{n}) \right\} \qquad (4)$$

If the horizon H is finite, then the optimal value function is obtained by computing $V^H(\vec{b}, \vec{x})$ and the optimal horizon-dependent policy $\pi^{*,h}$ at each stage h can be easily determined via $\pi^{*,h}(\vec{b}, \vec{x}) = \arg\max_a \arg\max_{\vec{y}} \arg\min_{\vec{n}} Q_a^h(\vec{b}, \vec{x}, \vec{y}, \vec{n})$. If the horizon $H = \infty$ and the optimal policy has finitely bounded value, then value iteration can terminate at horizon h if $V^h = V^{h-1}$; then $V^\infty = V^h$ and $\pi^{*,\infty} = \pi^{*,h}$.

Next we compute (3) and (4) using symbolic methods.

3 Symbolic Dynamic Programming (SDP)

In order to compute the equations above, we propose a *Robust symbolic dynamic programming* approach similar to [Sanner *et al.*, 2011]. This requires a general value iteration proposed in Algorithm 1 (`VI`) and a regression subroutine in Algorithm 2 (`Regress`) Before this RSDP approach we introduce the appropriate representation of *case* statements. All operations required for computing Q and V functions are *case operations* which are defined next.

Algorithm 1: `VI`(CSA-MDP, H) $\longrightarrow (V^h, \pi^{*,h})$

begin
- $V^0 := 0, h := 0$
- **while** $h < H$ **do**
  - $h := h + 1$
  - **foreach** $a(\vec{y}) \in A$ **do**
    - $Q_a^h(\vec{y}, \vec{n})$:= `Regress`$(V^{h-1}, a, \vec{y})$
    - $Q_a^h(\vec{y}) := \min_{\vec{n}} Q_a^h(\vec{y}, \vec{n})$ *//Stochastic* min
    - $Q_a^h := \max_{\vec{y}} Q_a^h(\vec{y})$ *// Continuous* max
    - $V^h :=$ casemax$_a$ Q_a^h *//* casemax *all* Q_a
    - $\pi^{*,h} := \arg\max_{(a,\vec{y})} Q_a^h(\vec{y})$
  - **if** $V^h = V^{h-1}$ **then**
    - break *// Terminate if early convergence*
- **return** $(V^h, \pi^{*,h})$

end

Algorithm 2: `Regress`$(V, a, \vec{y}) \longrightarrow Q$

1 **begin**
2 $Q =$ `Prime`(V) *// All* $b_i \rightarrow b_i'$ *and all* $x_i \rightarrow x_i'$
3 *// Continuous regression marginal integration*
4 **for** *all* x_j' *in* Q **do**
5 $Q := \int Q \otimes P(x_j'|\vec{b}, \vec{b}', \vec{x}, a, \vec{y}, \vec{n})\, d_{x_j'}$
6 *// Discrete regression marginal summation*
7 **for** *all* b_i' *in* Q **do**
8 $Q := \left[Q \otimes P(b_i'|\vec{b}, \vec{x}, a, \vec{y})\right]|_{b_i'=1}$
9 $\oplus \left[Q \otimes P(b_i'|\vec{b}, \vec{x}, a, \vec{y}, \vec{n})\right]|_{b_i'=0}$
10 $Q := R(\vec{b}, \vec{x}, a, \vec{y}) \oplus (\gamma \otimes Q)$
11 *//* max*-in noise variables*
12 **for** *all* n_k *in* Q **do**
13 $Q_a^h(\vec{y}, \vec{n}) :=$ casemax$_{n_k}$ $(Q, N(n_k, b_i', x_j'))$
14 **return** Q
15 **end**

3.1 Case Representation and Operators

Symbolic functions can generally be represented in *case* form [Boutilier *et al.*, 2001]:

$$f = \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases} \tag{5}$$

where ϕ_i are logical formulae defined over the state $(\vec{b}, \vec{x})$ and can include arbitrary logical $(\wedge, \vee, \neg)$ combinations of boolean variables and *linear* inequalities $(\geq, >, \leq, <)$ over continuous variables.

The f_i may be either linear or quadratic in the continuous parameters with no discontinuities at partition boundaries.

Unary operations such as scalar multiplication $c \cdot f$ and negation $-f$ on case statements f is performed by appling the operation to each f_i $(1 \leq i \leq k)$. *Binary operation* on two case statements are done by taking the cross-product of the logical partitions of the two case statement and performing the corresponding operation on the resulting paired partitions. The "cross-sum" $\oplus$ of two cases are defined as below:

$$\begin{cases} \phi_1 : & f_1 \\ \phi_2 : & f_2 \end{cases} \oplus \begin{cases} \psi_1 : & g_1 \\ \psi_2 : & g_2 \end{cases} = \begin{cases} \phi_1 \wedge \psi_1 : & f_1 + g_1 \\ \phi_1 \wedge \psi_2 : & f_1 + g_2 \\ \phi_2 \wedge \psi_1 : & f_2 + g_1 \\ \phi_2 \wedge \psi_2 : & f_2 + g_2 \end{cases}$$

The other operations of $\ominus$ and $\otimes$ are performed by subtracting or multiplying partition values. Note that some partitions may become inconsistent (infeasible) which are removed from the final result.

The other operations required for RSDP are restriction, substitution and maximization on case statements. `Regression` is defined separately for discrete and continuous variables.*Boolean restriction* $f|_{b=v}$ assigns the value $v \in \{0, 1\}$ to any occurrence of b in f. *Continuous integration* such as $\int Q(x_j') \otimes P(x_j'|\cdots)dx_j'$ in line 5 is performed symbolically to obtain $\int f(x_j') \otimes \delta[x_j' - h(\vec{z})]dx_j' = f(x_j')\{x_j'/h(\vec{z})\}$. Here $P(x_j'|\cdots)$ is in the form of $\delta[x_j' - h(\vec{z})]$ ($h(\vec{z})$ which is a case statement and $\vec{z}$ does not contain x_j'. Note that the latter operation in the result indicates that any occurrence of x_j' in $f(x_j')$ is *symbolically substituted* with the case statement $h(\vec{z})$. Full specification of these two operation is defined in [Sanner *et al.*, 2011].

A *symbolic case maximization* on two case statements is performed below:

$$\text{casemax}\left(\begin{cases} \phi_1 : f_1 \\ \phi_2 : f_2 \end{cases}, \begin{cases} \psi_1 : g_1 \\ \psi_2 : g_2 \end{cases} \right) = \begin{cases} \phi_1 \wedge \psi_1 \wedge f_1 > g_1 : f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \leq g_1 : g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2 : f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \leq g_2 : g_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 > g_1 : f_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 \leq g_1 : g_1 \\ \phi_2 \wedge \psi_2 \wedge f_2 > g_2 : f_2 \\ \phi_2 \wedge \psi_2 \wedge f_2 \leq g_2 : g_2 \end{cases}$$

If all f_i and g_i are linear, the casemax result is clearly still linear. If the f_i or g_i are quadratic (e.g. in the reward), then $f_i > g_i$ or $f_i \leq g_i$ will be at most univariate quadratic which can be linearized into two linear inequalities. Hence the result of this casemax operator will be representable in the case format with linear inequalities in decisions.

For continuous maximization according to **[?]** the $\max_y$ for each case partition is computed individually that is $\max_y \phi_i(\vec{b}, \vec{x}, y) f_i(\vec{b}, \vec{x}, y)$. In ϕ_i each conjoined constraint serves as either the upper bound on y, lower bound on y or independent of y. The casemax (casemin) operator is then used to determine the highest lower bound LB (lowest upper bound UB) for multiple symbolic upper and lower bounds on y. Apart from the lower and upper bound, the roots of $\frac{\partial}{\partial y} f_i$ w.r.t. y are also potential maxima points. These points are then symbolically evaluated to find which yields the maximizing value Max. Independent constraints and additional constraints that arise from the symbolic nature of the UB, LB, and $Root$ are also incorporated in the final result. To complete the maximization for an entire case statement f, the above procedure is applied to each case partition of f and the continuous maximization on f is the $\mathrm{casemax}$ of all individual results.

To implement the case statements efficiently with continuous variables, extended Algebraic Decision diagrams (XADDs) are used from [Sanner *et al.*, 2011] which is extended from ADDs [Bahar *et al.*, 1993]. Unreachable paths can be pruned in XADDs using LP solvers and all operations including the continuous minimization explained in the next section can be defined using XADDs. In the next section we show how to solve RDP symbolically using case statements.

3.2 Symbolic Robust Dynamic Programming

In this section we use the RCSA-MDP and case statement definitions to define the optimal value function V^h at horizon h as defined in section 2.2. Algorithm 1 (`VI`) outlines the main steps of a symbolic RDP approach.

Initially the value function V^h is assigned to 0. For every horizon the h-stage-to-go value functions $V^h(\vec{b},\vec{x})$ is computed as described in the following. For every action, the function Q_a^h is computed. Line 6 refers to Algorithm 2 which has the main steps below: (`Regress`) which has the following steps: (i) Substituting the next states in the value function. (ii) Performing Regression for continuous variable x in line 5 and boolean variable b in lines 8–9. (iii) Multiplying the regression by the discount factor and adding the reward function. (iii) Maximizing this result with the noise function in line 13. This step incorporates noise into the regressed Q-function consequently for each noise variable. Each noise variable assigns -∞ for legal values inside the boundary range +∞ for illegal values defined by the noise model $N(\vec{n},\vec{b},\vec{x})$.By maximizing in n_k all illegal values will remain +∞ since this is the maximum value compared to any other value and all legal values will be replaced by the regressed Q-value defined in step (iii) (-∞ is less than any other Q-value so it is omitted in the maximization).

Naturally a noisy process aims to minimize the noise in time to reach robustness Thus the regressed stochastic $Q_a^h(\vec{y},\vec{n})$ is now minimized over the noise variables $\vec{n}$ in line 7. Intuitively this continuous minimization will never choose +∞ as there is always some value smaller which insures that the transitioned model never chooses illegal values. All legal Q-values are considered in the minimization step to find the value corresponding to the minimum noise. This continuous minimization is similar to that of continuous maximization explained in the previous section. The minima points of upper and lower bounds on n_k are evaluated for the minimum possible value of noise. The $LB \leq UB$ constraint and the constraints independent of n_k are also considered for the minimization on a single partition. The final result is a $\mathrm{casemin}$ on all the individual minimum results.

The resulting Q-value with minimal noise is maximized over the continuous action parameter in line 8; a symbolic continuous maximization operation. A discrete $\mathrm{casemax}$ on the set of discrete actions for all Q-functions defines the final V and the optimal policy is defined as the $\arg\max$ over the set of discrete and continuous actions on Q.

4 Empirical Results

5 Related Work

Hybrid systems are a class of dynamical systems that involve both continuous and discrete dynamics. The dynamics of the continuous variables are defined typically through differential equations and the evolution of the discrete variables through finite state machines, Petri nets or other abstract computational machines. One accepted manner to model hybrid systems is using hybrid automata that represents, in a single formalism, the discrete changes by automata transitions and the continuous changes by differential equations [De Schutter *et al.*, 2009; Henzinger *et al.*, 1997]. One special class of hybrid systems are the switched linear systems that have a collection of subsystems defined by linear dynamics (differential equations) and a switching rule that specifies the switching between the subsystems [Sun and Ge, 2005].

One problem that has been studied in the area of hybrid systems is the verification of the safety property, that tries to proof that the system does not enter in unsafe configurations from an initial configuration [Tomlin *et al.*, 2003]. Then, we say that the system satisfies the safety property if all reachable states are safe [Henzinger *et al.*, 1997]. There are many tools for the automatic verification of hybrid systems such as HyTech [Henzinger *et al.*, 1997], KRONOS [Yovine, 1997], PHAVer [Frehse, 2005] and HSOLVER [Ratschan and She, 2007]. All the techniques rely on the ability to compute reachable sets of hybrid systems. For example, HyTech, a symbolic model checker, automatically computes reachable sets for linear hybrid automata, a subclass of hybrid automata. HyTech can also return the values of design parameters for which this automata satisfies a temporal-logic requirement [Henzinger *et al.*, 1997]. Some examples of verification of hybrid systems can be found in [Henzinger *et al.*, 1997; von Mohrenschildt, 2001].

Another challenging topic in hybrid systems is to evaluate the effect of the hybrid controller on the systems operation, i.e., to solve the controllability problem for hybrid systems [Stikkel *et al.*, 2004]. A hybrid system is called hybrid controllable if, for any pair of valid states, there exists at least one permitted control sequence (correct control-laws) between them [Tittus and Egardt, 1998; Yang and Blanke, 2007]. The general controllability problem of hybrid systems is NP hard [Blondel and Tsitsiklis, 1999]. However, for special classes of hybrid systems, some necessary and sufficient conditions for controllability were obtained in [Stikkel *et al.*, 2004; Lemch *et al.*, 2001; Sun *et al.*, 2002; Yang, 2003; Yang and Blanke, 2007]. For example, by employing algebraic manipulation of system matrices, a sufficient and necessary condition for the controllability analysis of a class of piecewise linear hybrid systems is given in [Yang, 2003]. This class is called controlled switching linear hybrid system and have the following properties: all mode switches are controllable, the dynamical subsystems within each mode has a LTI form, the admissible operating regions within each mode is the whole state space, and there are no discontinuous state jumps. The controllability test for this class of hybrid system can be determined based on the system matrices. In [Yang and Blanke, 2007] is proposed an approach for controllability analysis of a class of more complex hybrid systems. This approach uses a discrete-path searching algorithm that integrates global reachability analysis at the discrete event system level and a local reachability analysis at the continuous level. This method cannot guarantee the existence of a solution for an arbitrary hybrid system [Yang and Blanke, 2007].

Much of the work on hybrid systems has focused on deterministic models without allowing any uncertainty. In practice, there are real world applications where the environment is inherent uncertainty. To cope with this, the stochastic hybrid systems was proposed. Stochastic hybrid systems allow uncertainty (1) replacing deterministic jumps between discrete states by random jumps or (2) replacing the deterministic dynamics inside the discrete state by a stochastic differential equation or (3) combinations of 1 and 2 [Hu *et al.*, 2000]. A critical problem in this type of systems is the verification of reachability properties because it is necessary to cope with the interaction between the discrete and continuous stochastic dynamics, in this case it is computed the probability that the system satisfies the property [Koutsoukos and Riley, 2006]. Related with the concept of verification of safety property, in stochastic hybrid systems, the system tries to maximize the probability that the execution will remain in safe states as long as possible [Hu *et al.*, 2000].

Chance-constrained predictive stochastic control of dynamic systems characterizes uncertainty in a probabilistic manner, and finds the optimal sequence of control inputs subject to the constraint that the probability of failure must be below a user-specified threshold [Blackmore *et al.*, 2011]. This constraint is known as a chance constraint [Blackmore *et al.*, 2011] and is used to define stochastic robustness.

A great deal of work has taken place in recent years relating to chance-constrained optimal control of linear systems subject to Gaussian uncertainty in convex regions [Schwarm and Nikolaou, 1999; Li *et al.*, 2002; Ono and Williams, 2008] and linear systems in nonconvex regions [Blackmore *et al.*, 2010; 2011]. The approach in [Blackmore *et al.*, 2010] uses samples, or particles, to approximate the chance constraint, and hence does not guarantee satisfaction of the constraint. It applies to arbitrary uncertainty distributions and is significantly more computationally intensive than others. The approach proposed in [Blackmore *et al.*, 2011] uses analytic bound to ensure satisfaction of the constraint and applies for linear-Gaussian systems.

6 Concluding Remarks

This work has combined symbolic techniques and data structures from the HMDP literature in AI with techniques from chance-constrained control theory to provide optimal robust solutions to a range of problems with general continuous transitions and state-dependent noise for which no general exact closed-form solutions previously existed. Using these techniques we were able to find optimal policies and answer questions of robust controllability for a variety of highly risk-sensitive applications from AI planning, control theory, and operations research such as UAV NAVIGATION, SPACE TELESCOPE CONTROL, and RESERVOIR CONTROL. Among many potential avenues for future work, combining this receding horizon control approach with focused search techniques as in HAO* [Meuleau *et al.*, 2009] should preserve our strong robust optimality guarantees while substantially increasing the scalability of our approach in exchange for restricting solution optimality to a known set of initial states.

References

[Bahar *et al.*, 1993] R. Iris Bahar, Erica Frohm, Charles Gaona, Gary Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. Algebraic Decision Diagrams and their applications. In *IEEE /ACM International Conference on CAD*, 1993.

[Bellman, 1957] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[Blackmore *et al.*, 2010] L. Blackmore, M. Ono, A. Bektassov, and B.C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *Robotics, IEEE Transactions on*, 26(3):502 –517, 2010.

[Blackmore *et al.*, 2011] Lars Blackmore, Masahiro Ono, and Brian C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094, 2011.

[Blondel and Tsitsiklis, 1999] Vincent D. Blondel and John N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3):479–489, 1999.

[Boutilier *et al.*, 2001] Craig Boutilier, Ray Reiter, and Bob Price. Symbolic dynamic programming for first-order MDPs. In *IJCAI-01*, pages 690–697, Seattle, 2001.

[Boyan and Littman, 2001] Justin Boyan and Michael Littman. Exact solutions to time-dependent MDPs. In *Advances in Neural Information Processing Systems NIPS-00*, pages 1026–1032, 2001.

[De Schutter *et al.*, 2009] B. De Schutter, W.P.M.H. Heemels, J. Lunze, and C. Prieur. Survey of modeling, analysis, and control of hybrid systems. In J. Lunze and F. Lamnabhi-Lagarrigue, editors, *Handbook of Hybrid Systems Control – Theory, Tools, Applications*, chapter 2, pages 31–55. Cambridge University Press, Cambridge, UK, 2009.

[Feng *et al.*, 2004] Zhengzhu Feng, Richard Dearden, Nicolas Meuleau, and Richard Washington. Dynamic programming for structured continuous markov decision problems. In *Uncertainty in Artificial Intelligence (UAI-04)*, pages 154–161, 2004.

[Frehse, 2005] Goran Frehse. Phaver: Algorithmic verification of hybrid systems past hytech. In *Hybrid Systems: Computation and Control International Workshop*, pages 258–273, 2005.

[Henzinger *et al.*, 1997] Thomas A. Henzinger, Pei H. Ho, and Howard W. Toi. HYTECH: A Model Checker for Hybrid Systems. *International Journal on Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.

[Hu *et al.*, 2000] J. Hu, John Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. *Lecture Notes in Computer Science LNCS*, 1790:160–173, 2000.

[Koutsoukos and Riley, 2006] Xenofon Koutsoukos and Derek Riley. Computational methods for reachability analysis of stochastic hybrid systems. In *Proceedings of the 9th international conference on Hybrid Systems: computation and control*, HSCC'06, pages 377–391, Berlin, Heidelberg, 2006. Springer-Verlag.

[Kveton *et al.*, 2006] Branislav Kveton, Milos Hauskrecht, and Carlos Guestrin. Solving factored mdps with hybrid state and action variables. *Journal Artificial Intelligence Research (JAIR)*, 27:153–201, 2006.

[Lemch *et al.*, 2001] Ekaterina S. Lemch, Shankar Sastry, and Peter E. Caines. Global controllability of hybrid systems with controlled and autonomous switchings. In *Hybrid Systems: Computation and Control International Workshop*, pages 375–386, 2001.

[Li and Littman, 2005] Lihong Li and Michael L. Littman. Lazy approximation for solving continuous finite-horizon mdps. In *National Conference on Artificial Intelligence AAAI-05*, pages 1175–1180, 2005.

[Li *et al.*, 2002] Pu Li, Moritz Wendt, and GüNter Wozny. Brief a probabilistically constrained model predictive controller. *Automatica*, 38(7):1171–1176, 2002.

[Marecki *et al.*, 2007] Janusz Marecki, Sven Koenig, and Milind Tambe. A fast analytical algorithm for solving markov decision processes with real-valued resources. In *International Conference on Uncertainty in Artificial Intelligence IJCAI*, pages 2536–2541, 2007.

[Meuleau *et al.*, 2009] Nicolas Meuleau, Emmanuel Benazera, Ronen I. Brafman, Eric A. Hansen, and Mausam. A heuristic search approach to planning with continuous resources in stochastic domains. *Journal Artificial Intelligence Research (JAIR)*, 34:27–59, 2009.

[Ono and Williams, 2008] Masahiro Ono and Brian C. Williams. An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. In *AAAI*, pages 1376–1382, 2008.

[Puterman, 1994] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

[Ratschan and She, 2007] Stefan Ratschan and Zhikun She. Safety verification of hybrid systems by constraint propagation-based abstraction refinement. *ACM Trans. Embed. Comput. Syst.*, 6(1), 2007.

[Sanner *et al.*, 2011] Scott Sanner, Karina Valdivia Delgado, and Leliane Nunes de Barros. Symbolic dynamic programming for discrete and continuous state mdps. In *Proceedings of the 27th Conference on Uncertainty in AI (UAI-2011)*, Barcelona, 2011.

[Schwarm and Nikolaou, 1999] Alexander T. Schwarm and Michael Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45(8):1743–1752, 1999.

[Stikkel *et al.*, 2004] Gábor Stikkel, Jozsef Bokor, and Zoltán Szabó. Necessary and sufficient condition for the controllability of switching linear hybrid systems. *Automatica*, 40(6):1093–1097, 2004.

[Sun and Ge, 2005] Zhendong Sun and Shuzhi Sam Ge. Analysis and synthesis of switched linear control systems. *Automatica*, 41(2):181–195, 2005.

[Sun *et al.*, 2002] Zhendong Sun, Shuzhi Sam Ge, and Tong Heng Lee. Controllability and reachability criteria for switched linear systems. *Automatica*, 38(5):775–786, 2002.

[Tittus and Egardt, 1998] M. Tittus and B. Egardt. Control design for integrator hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):491 –500, apr 1998.

[Tomlin *et al.*, 2003] Claire Tomlin, Ian Mitchell, Alexandre M. Bayen, and Meeko Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[von Mohrenschildt, 2001] Martin von Mohrenschildt. Symbolic verification of hybrid systems: An algebraic approach. *European Journal of Control*, 7(5):541–556, 2001.

[Yang and Blanke, 2007] Zhenyu Yang and Mogens Blanke. A unified approach to controllability analysis for hybrid control systems. *Nonlinear Analysis: Hybrid Systems*, 1(2):212 – 222, 2007.

[Yang, 2003] Zhenyu Yang. A sufficient and necessary condition for the controllability of linear hybrid systems. In *IEEE International Symposium on Intelligent Control*, pages 128 –133, oct. 2003.

[Yovine, 1997] Sergio Yovine. Kronos: A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*, 1:123–133, 1997.

[Zamani *et al.*, 2012] Z. Zamani, S. Sanner, and C. Fang. Symbolic dynamic programming for continuous state and action mdps. In *In Proceedings of the 26th AAAI Conference on Artificial Intelligence (***AAAI-12***)*, Toronto, Canada, 2012.