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 pre-specified 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; Zamani *et al.*, 2012] 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 w.r.t. constrained non-deterministic transitions in HMDPs. This permits us to find optimal robust solutions for a wide range of non-deterministic HMDPs and allows 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 Non-deterministic Hybrid MDPs

We first formally introduce the framework of Hybrid (discrete and continuous) Markov decision processes with non-deterministic continuous noise (ND-HMDPs) by extending the HMDP framework of [Zamani *et al.*, 2012]. The optimal solution for this model is then defined via robust dynamic programming.

2.1 Factored Representation

An HMDP 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 a discrete boolean variable and each $x_j \in \mathbb{R}$ $(1 \le j \le c)$ is continuous. To model continuous uncertainty in ND-HMDPs we additionally define intermediate noise variables $\vec{n} = n_1,\ldots,n_e$ where each $n_l \in \mathbb{R}$ $(1 \le l \le e)$. Both discrete and continuous actions are represented in the set $A = \{a_1(\vec{y}_1),\ldots,a_p(\vec{y}_p)\}$ where each action $a(\vec{y}) \in A$ references a (possibly empty) vector of continuous parameters $\vec{y} \in \mathbb{R}^{|\vec{y}|}$; we say an action is discrete if it has no continuous parameters ($|\vec{y}| = 0$), otherwise it is continuous.

Given a current state $(\vec{b},\vec{x})$ and next state $(\vec{b}',\vec{x}')$ and an executed action $a(\vec{y})$ at the current state, a real-valued reward function $R(\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y})$ specifies the immediate reward obtained at the current 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. In a factored setting, we do not typically represent the transition distribution jointly but rather we factorize it into a dynamic Bayes net (DBN) [] as follows:

$$P(\vec{b}',\vec{x}'|\vec{b},\vec{x},a,\vec{y},\vec{n}) = \prod_{i=1}^{a} P(b_i'|\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y},\vec{n}) \prod_{j=1}^{c} P(x_j'|\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y},\vec{n}) \quad (1)$$

Here we allow synchronic arcs under the condition that the DBN forms a proper directed acyclic graph (DAG). For binary variables b_i $(1 \le i \le a)$, $P(b_i'|\vec{b},\vec{x},\vec{b}',\vec{x}',\vec{x},a,\vec{y},\vec{n})$ are defined as general conditional probability functions (CPFs), which are not necessarily tabular since they may condition on inequalities over continuous variables. For continuous variables x_j $(1 \le j \le c)$, the CPFs $P(x_j'|\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y},\vec{n})$ are represented with *piecewise linear equations* (PLEs) that may have piecewise conditions which are arbitrary logical combinations of $\vec{b}$, $\vec{b}'$ and linear inequalities over $\vec{x}$, $\vec{x}'$, and $\vec{n}$. Examples of PLEs will follow shortly.

In general, we assume that for each intermediate continuous noise variable n_l $(1 \le l \le e)$ a non-deterministic noise interval constraint function $N(n_l|\vec{b},\vec{x},a,\vec{y})$ has been defined that represents a range covering α of the probability mass for n_l and evaluates to $-\infty$ for legal values of n_l and $+\infty$ otherwise. The reason for the $\pm\infty$ evaluation is simple: in a robust solution to HMDPs with non-deterministic noise constraints, Nature will attempt to adversarially minimize the reward the agent can achieve and hence we let $N(n_l|\vec{b},\vec{x},a,\vec{y})$ take the value $+\infty$ for illegal values of n_l to ensure Nature will never choose illegal assignments of n_l when minimizing.

As an intuitive example, if $P(n_l|\vec{b},\vec{x},a,\vec{y}) = \mathcal{N}(n_l;\mu;\sigma^2)$ is a simple Normal distribution with mean μ and variance σ^2 and we let $\alpha = 0.95$ then we know that that the 95% of the probability mass lies within $\mu \pm 2\sigma$, hence

$$N(n_l|\vec{b},\vec{x},a,\vec{y}) = \begin{cases} \mu - 2\sigma \le n_l \le \mu + 2\sigma : & -\infty \\ \text{otherwise} : & +\infty \end{cases} .$$

To make the ND-HMDP framework concrete, we now introduce a running example used throughout the paper:

Example (RESERVOIR CONTROL [?]). **ZAHRA TODO: enter definition and explain**

We formally define this ND-HMDP using the following piecewise dynamics and reward: **ZAHRA TODO: no need to specify binary variables – just say a counter to indicate day of week, but specify continuous CPFs/PLEs and noise constraints** Here use of the $\delta[\cdot]$ function ensures that the continuous CPF over x' integrates to 1.

A policy $\pi(\vec{b},\vec{x})$ specifies the action $a(\vec{y}) = \pi(\vec{b},\vec{x})$ to take at state $(\vec{b},\vec{x})$. In a robust solution to HMDPs with non-deterministic noise constraints, 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 γ, $0 \le \gamma \le 1$, the expected sum of discounted rewards over horizon $h \in H$ $(H \ge 0)$ is maximized subject to Nature's adversarial attempt to choose value minimizing assignments of the noise variables. The value function V w.r.t. Π^* in this case is defined via a recursive expectation

$$V^{\Pi^{*,H}}(\vec{b},\vec{x}) = \min_{\vec{n}} \max\Big(N(n_1|\vec{b},\vec{x},\Pi^{*,H}),\ldots,$$
$$\max\Big(N(n_e|\vec{b},\vec{x},\Pi^{*,H}), E_{\Pi^{*},H}\Big[r^h + \gamma V^{\Pi^{*,H-1}}(\vec{b}',\vec{x}')\Big|\vec{b}_0,\vec{x}_0\Big]\Big)\cdots\Big)$$

where r^h is the reward obtained at horizon h following policy Π^* and using Nature's minimizing choice of $\vec{n}$ at each h.

The effect of "max'ing" in each of the previously defined $N(n_l|\vec{b},\vec{x},a,\vec{y})$ $(1 \le l \le e)$ with the value function is one of the major insights and contributions of this paper. We noted before that Nature will never choose an illegal value of n_l where $N(n_l|\vec{b},\vec{x},a,\vec{y}) = +\infty$, instead it will choose a legal value of n_l for which $N(n_l|\vec{b},\vec{x},a,\vec{y}) = -\infty$ which when "max'ed" in with the value function effectively vanishes owing to the indentity $\max(v,-\infty) = v$ for all $v > -\infty$.

Finally, by leveraging the simple union bound, we can easily prove that that a policy will achieve $V^{\Pi^{*,H}}$ with at least $1-H(1-\alpha)$ probability since the probability of encountering a noise value outside the confidence interval is only $(1-\alpha)$ at any time step. Hence for a success probability of at least β, one should choose $\alpha = 1 - \frac{1-\beta}{H}$, e.g., $\beta = 0.95$ success probability requires an $\alpha = 0.99$ for $H = 5$.

2.2 Robust Dynamic Programming

We extend the value iteration dynamic programming algorithm [Bellman, 1957] and specifically the form used for HMDPs in [Zamani *et al.*, 2012] to a robust dynamic programming (RDP) algorithm for ND-HMDPs that may be considered a continuous action generalization of zero-sum alternating turn Markov games [Littman, 1994]. Initializing

$V^0(\vec{b},\vec{x}) = 0$ the algorithm builds the h-stage-to-go value function $V^h(\vec{b},\vec{x})$.

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

$$Q_a^h(\vec{b},\vec{x},\vec{y},\vec{n}) = \max\Big(N(n_1|\vec{b},\vec{x},\Pi^{*,H}),\ldots,\max\Big(N(n_e|\vec{b},\vec{x},\Pi^{*,H}),$$
$$\sum_{\vec{b}'}\int\prod_{i=1}^{a}P(b_i'|\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y},\vec{n})\prod_{j=1}^{c}P(x_j'|\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y},\vec{n})$$
$$\Big[R(\vec{b},\vec{x},\vec{b}',\vec{x}',a,\vec{y}) + \gamma V^{h-1}(\vec{b}',\vec{x}')d\vec{x}'\Big]\Big)\cdots\Big) \quad (2)$$

Here the noise constraints $N(\vec{n},\vec{b},\vec{x})$ are "max'ed" in with the value function to ensure Nature chooses a legal setting of n_l, effectively reducing each $\max$ to an identity operation.

Next, given $Q_a^h(\vec{b},\vec{x},\vec{y},\vec{n})$ as above for each $a \in A$, we can proceed to define the h-stage-to-go value function assuming that the agent attempts to maximize value subject to Nature's adversarial choice of value-minimizing noise:

$$V^h(\vec{b},\vec{x}) = \max_{a\in A}\max_{\vec{y}\in\mathbb{R}^{|\vec{y}|}}\min_{\vec{n}\mathbb{R}^{|\vec{e}|}}\left\{Q_a^h(\vec{b},\vec{x},\vec{y},\vec{n})\right\} \quad (3)$$

The optimal policy at horizon h can also be determined using the Q-function as below:

$$\pi^{*,h}(\vec{b},\vec{x}) = \arg\max_{a\in A}\arg\max_{\vec{y}\in\mathbb{R}^{|\vec{y}|}}\min_{\vec{n}\mathbb{R}^{|\vec{e}|}} Q_a^h(\vec{b},\vec{x},\vec{y},\vec{n}) \quad (4)$$

For finite-horizon HMDPs the optimal value function and policy are obtained up to horizon H. For infinite horizons where the optimal policy has finitely bounded value then value iteration terminates when two values are equal in subsequent horizons ($V^h = V^{h-1}$). In this case $V^\infty = V^h$ and $\pi^{*,\infty} = \pi^{*,h}$.

Up to this point we have only provided the abstract mathematical framework for ND-HMDPs and RDP. Fortuitously though, we can leverage the continuous $\max$ (and analogously defined $\min$) operations and symbolic DP approach of [Zamani *et al.*, 2012] in order to compute RDP via (2) and (3) exactly in closed-form. We discuss this next.

3 Robust Symbolic Dynamic Programming

In order to compute the equations above, we propose a *robust symbolic dynamic programming* (RSDP) approach building on the work of [**?**; Sanner *et al.*, 2011]. This requires a value iteration algorithm described in Algorithm 1 (VI) and the regression subroutine described in Algorithm 2. In what follows we show how the techniques of SDP can be extended to compute RDP exactly in closed-form as discussed in the last section.

In general we define *all* symbolic functions to be represented in *case* form [Boutilier *et al.*, 2001] for which a binary "cross-sum" operation can be defined as follows:

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

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

```
1  begin
2      $V^0 := 0, h := 0$
3      while $h < H$ do
4          $h := h + 1$
5          foreach $a(\vec{y}) \in A$ do
6              $Q_a^h(\vec{y},\vec{n})$ := Regress($V^{h-1}, a, \vec{y}$)
7              $Q_a^h(\vec{y}) := \min_{\vec{n}} Q_a^h(\vec{y},\vec{n})$ //Stochastic min
8              $Q_a^h := \max_{\vec{y}} Q_a^h(\vec{y})$ // Continuous max
9              $V^h := \text{casemax}_a Q_a^h$ // casemax all $Q_a$
10             $\pi^{*,h} := \arg\max_{(a,\vec{y})} Q_a^h(\vec{y})$
11         if $V^h = V^{h-1}$ then
12             break // Terminate if early convergence
13
14     return $(V^h, \pi^{*,h})$
15 end
```

Here ϕ_i and ψ_j 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 – we call this *linear case form* (LCF). The f_i and g_j are restricted to be *linear* functions. Similarly operations such as $\ominus$ and $\otimes$ may be defined with operations applied to LCF functions yielded LCF results.

ZAHRA TODO: define casemax here, then introduce remaining operations as used to compute results for Reservoir running example getting from V^0 to V^1... showing whatever you feel is most important by the end of page 4

NOTE: need to rewrite a lot this section from scratch... try not to be too verbose... we've already said why we do each operation so you just have to remark on the running example and what the result of each iteration says... don't explain the operations again unless something is really non-obvious... assume the reader can easily figure out the mechanical details of the derivation

While explaining the steps of the VI algorithm we show that all operations required to compute the optimal policy is supported by case representation and defined symbolically.

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. To follow the steps, we use the second iteration of the Rover example here. For simplification, we omit the boolean variable b of taking a picture and only use one noisy continuous variable x. We now perform steps 1–4 for $h = 2$.

(1) For every action, the function Q_a^h is computed. Line 6 refers to Algorithm 2 which has the main steps below:

(1a) Priming the current state variables (b_i, x_j) to build the next states (b_i', x_j') in the value function. This indicates that any occurrence of the current state in V^h is *symbolically substituted* with the next state variables that is $V'^h = V^h\sigma$ where $\sigma = \{b_i \setminus b_i', x_j \setminus x_j'\}$ for all values of i and j. For the second iteration this step is equal to the following:

```
Algorithm 2: Regress($V, a, \vec{y}$) ⟶ $Q$
begin
    $Q = $ Prime($V$)  // All $b_i \to b'_i$ and all $x_i \to x'_i$
    ZAHRA TODO: Need to change a little now
    that we allow synchronic arcs... change to
    for all $v' \in Q$
    child variables must come parents in order
    if ($v' = x'_j$) then do continuous regression
    else if ($v' = b'_i$) then do discrete regression
    ... just need to reformat what is below to fit this
    new control flow  // Continuous regression
    marginal integration
    for all $x'_j$ in $Q$ do
        $Q := \int Q \otimes P(x'_j|\vec{b}, \vec{b}', \vec{x}, a, \vec{y}, \vec{n})\, d_{x'_j}$
    // Discrete regression marginal summation
    for all $b'_i$ in $Q$ do
        $Q := \left[Q \otimes P(b'_i|\vec{b}, \vec{x}, a, \vec{y}, \vec{n})\right]|_{b'_i=1}$
                $\oplus \left[Q \otimes P(b'_i|\vec{b}, \vec{x}, a, \vec{y}, \vec{n})\right]|_{b'_i=0}$
    $Q := R(\vec{b}, \vec{x}, a, \vec{y}) \oplus (\gamma \cdot Q)$
    // max-in noise variables
    for all $n_k$ in $Q$ do
        $Q^h_a(\vec{y}, \vec{n}) := \mathrm{casemax}_{n_k}\,(Q, N(n_k, b'_i, x'_j))$
    return $Q$
end
```

$$Q = V'^1 = \begin{cases} -40 \le x' \le 40 : & 10 \\ x > 40 \vee x' < -40 : & -\infty \end{cases}$$

(1b) Performing Regression for continuous variable x in line 5 and boolean variable b in lines 8–9. *Boolean restriction* $f|_{b=v}$ assigns the value $v \in \{0, 1\}$ to any occurrence of b in f.

Continuous integration of $\int Q(x'_j) \otimes P(x'_j|\cdots)dx'_j$ where results in the following according to the rules of integration:

$$\int f(x'_j) \otimes \delta[x'_j - h(\vec{z})]dx'_j = f(x'_j)\{x'_j/h(\vec{z})\}$$

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). The latter operation in the result indicates that any occurrence of x'_j in $f(x'_j)$ is substituted with the case statement $h(\vec{z})$. For our example this step results in the following intermediate Q-value:

$$\begin{cases} -40 \le x' \le 40 : & 10 \\ x > 40 \vee x' < -40 : & -\infty \end{cases} \otimes \delta\left(x' - (x + n + a)\right) =$$
$$\begin{cases} -40 \le (x + a + n) \le 40 : & 10 \\ (x + n + a) > 40 \vee (x + n + a) < -40 : & -\infty \end{cases}$$

(1c) Multiplying the regression by the discount factor and adding the reward function in line 10. *Unary operations* such as scalar multiplication $\gamma \cdot Q$ (and also negation $-Q$) on case statements Q is performed by applying the operation to each Q_i ($1 \le i \le k$) while adding the reward is a binary $\oplus$:

$$Q = V'^1 = \begin{cases} -40 \le (x + n + a) \le 40 : & 20 \\ (x + n + a) > 40 \vee (x + n + a) < -40 : & -\infty \\ x > 40 \vee x < -40 : & -\infty \end{cases}$$

(1d) 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 (1c) -∞ is less than any other Q-value so it is omitted in the maximization. The Rover example is redefined with this noise variable as below:

$$Q = \begin{cases} -5 \le n \le 5 : & +\infty \\ (-40 \le (x + n + a) \le 40) \wedge \neg(-5 \le n \le 5) : & 20 \\ \neg(-40 \le (x + n + a) \le 40) \wedge \neg(-5 \le n \le 5) : & -\infty \\ (x > 40 \vee x < -40) \wedge (-5 \le n \le 5) : & +\infty \\ (x > 40 \vee x < -40) \wedge \neg(-5 \le n \le 5) : & -\infty \end{cases}$$

(2) Naturally a noisy process aims to minimize the noise to reach robustness Thus the regressed stochastic $Q^h_a(\vec{y}, \vec{n})$ from Algorithm 2 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. Each partition i of this intermediate Q is considered for a continuous minimization separately with the final result a casemin on all the individual minimum results $\mathrm{casemin}_i \min_n \phi_i(\vec{b}, \vec{x}, \vec{n}) f_i(\vec{b}, \vec{x}, \vec{n})$. We demonstrate the steps of this algorithm for the second partition of Q defined as:

$$\min_n (-40 \le (x + n + a) \le 40) \wedge \neg(-5 \le n \le 5) : 20$$

For each partition the logical constraints are used to derive the (a) lower bound on n (LB = -5, $-40 - a - x$); (b) upper bound on n (UB = 5, $40 - a - x$) and (c) constraints independent of n (IND). In case of several bounds on n the maximum of all lower bounds and the minimum of all upper bounds is desired. A *symbolic case maximization* on two case statements of ($\phi_i : f_i$) and (ψ_i, g_i) where ($i \in \{1, 2\}$) is performed below.

$$\mathrm{casemax} = \begin{cases} \phi_1 \wedge \psi_1 \wedge f_1 > g_1 : f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \le g_1 : g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2 : f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \le g_2 : g_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 > g_1 : f_2 \\ \phi_2 \wedge \psi_1 \wedge f_2 \le g_1 : g_1 \\ \phi_2 \wedge \psi_2 \wedge f_2 > g_2 : f_2 \\ \phi_2 \wedge \psi_2 \wedge f_2 \le g_2 : g_2 \end{cases}$$

Thus the bounds are defined as below:

$$LB = \begin{cases} a + x < -35 : & -40 - x - a \\ a + x \geq -35 : & -5 \end{cases}$$
$$UB = \begin{cases} a + x < 35 : & 5 \\ a + x \geq -35 : & 40 - a - x \end{cases}$$

The minima points of upper and lower bounds are evaluated for the leaf value which equals to substituting the bounds instead of the noise variable n in the leaf function. The minimum of these two evaluations are then stored, note that in our example the leaf is a constant 20 value which is not effected by this step.

Natural constraints on bounds $LB \leq UB$ and the IND constraints are also considered for the minimization on a single partition to obtain:

$$Q = \begin{cases} (-40 \leq x \leq 40) \wedge (-45 \leq (x + a) \leq 45) : & 20 \\ (-40 \leq x \leq 40) \wedge \neg(-45 \leq (x + a) \leq 45) : & +\infty \\ (x > 40 \vee x < -40) : & +\infty \end{cases}$$

The final result of a continuous minimization is a casemin over all partitions which results in the following Q-value:

$$Q = \begin{cases} (-40 \leq x \leq 40) \wedge (-35 \leq (x + a) \leq 35) : & 20 \\ (-40 \leq x \leq 40) \wedge \neg(-35 \leq (x + a) \leq 35) : & -\infty \\ (x > 40 \vee x < -40) : & -\infty \end{cases}$$

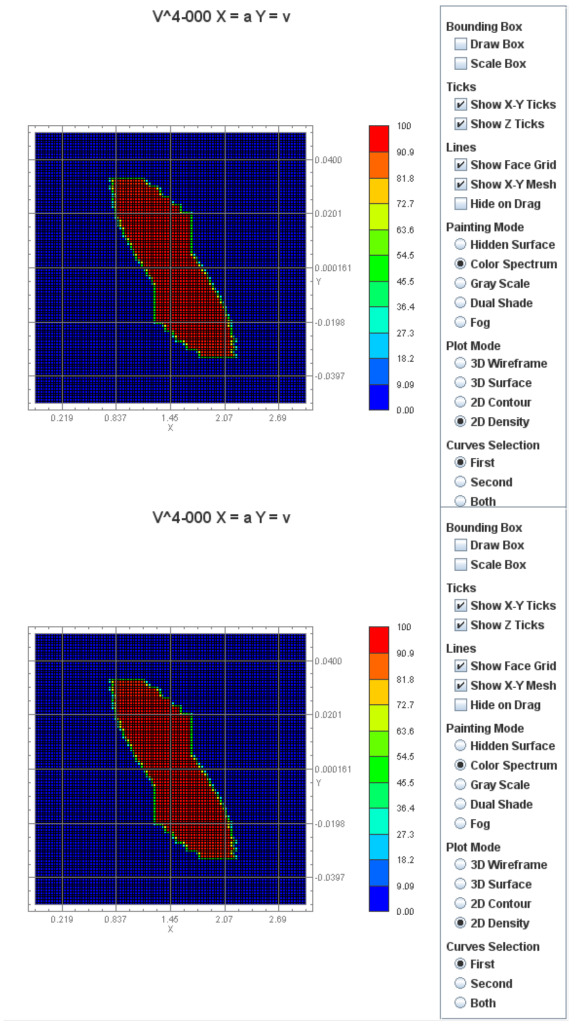
3 The resulting Q-value with minimal noise is maximized over the continuous action parameter in line 8; a symbolic continuous maximization operation, the major contribution of [Zamani *et al.*, 2012]. The resulting Q_a^h can be determined as the following:

$$Q = \begin{cases} (-40 \leq x \leq 40) : & 20 \\ (x > 40 \vee x < -40) : & -\infty \end{cases}$$

4 A discrete 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. In our example the final value $V^h = Q^h$ because there is only one single discrete action.

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 can be defined using XADDs by treating each path from root to leaf node as a single case partition with conjunctive constraints, $\min_n$ is performed at each leaf subject to these constraints and all path $\min_n$'s are then accumulated via the casemin operation to obtain the final result.

4 Empirical Results



Bounding Box
☐ Draw Box
☐ Scale Box
Ticks
☑ Show X-Y Ticks
☑ Show Z Ticks
Lines
☑ Show Face Grid
☑ Show X-Y Mesh
☐ Hide on Drag
Painting Mode
○ Hidden Surface
● Color Spectrum
○ Gray Scale
○ Dual Shade
○ Fog
Plot Mode
○ 3D Wireframe
○ 3D Surface
○ 2D Contour
● 2D Density
Curves Selection
● First
○ Second
○ Both

Bounding Box
☐ Draw Box
☐ Scale Box
Ticks
☑ Show X-Y Ticks
☑ Show Z Ticks
Lines
☑ Show Face Grid
☑ Show X-Y Mesh
☐ Hide on Drag
Painting Mode
○ Hidden Surface
● Color Spectrum
○ Gray Scale
○ Dual Shade
○ Fog
Plot Mode
○ 3D Wireframe
○ 3D Surface
○ 2D Contour
● 2D Density
Curves Selection
● First
○ Second
○ Both

Figure 2: Space and elapsed time (between current and previous horizon) vs. horizon.

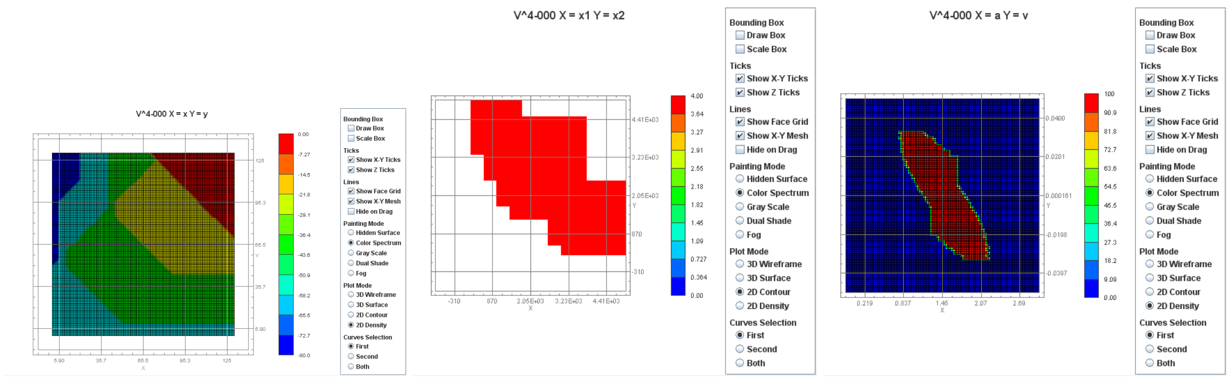


Figure 1: *(left)* $V^4(x, y)$UAV NAVIGATION problem; *(middle)* $V^4(l_1, l_2)$ RESERVOIR CONTROL problem; *(right)* $V^4(a, v)$ SPACE TELESCOPE CONTROL problem.

We evaluated RH-MDP using XADDs on UAV NAVIGATION problem; RESERVOIR CONTROL problem and SPACE TELESCOPE CONTROL problem — described below.[1]

UAV NAVIGATION: The state consist of UAVs continuous position x and y. In a given time step, the UAV may move a continuous distance $ax \in [-40, 40]$ and $ay \in [-40, 40]$. The turbulence introduces a noise nx and ny respectively in the movement, given by:

$$nx = \begin{cases} (y \geq 50 + x) \wedge (nx \leq -20) \wedge (nx \geq 20) & : legal \\ (y < 50 + x) \wedge (nx \leq -5) \wedge (nx \geq 5) & : legal \\ else & : illegal \end{cases}$$

$$ny = \begin{cases} (y \geq 50 + x) \wedge (ny \leq -20) \wedge (ny \geq 20) & : legal \\ (y < 50 + x) \wedge (ny \leq -5) \wedge (ny \geq 5) & : legal \\ else & : illegal \end{cases}$$

The UAV goal is to achieve the region $x + y > 200$. It receives a reward penalty $(-\infty)$ for being in positions from which a UAV with a given amount of fuel reserves cannot return to its landing strip and, if the UAV is not in the goal position $(\neg l)$, the action cost is 20:

$$R = \begin{cases} (l) \wedge (x \leq 130) \wedge (y \leq 130) \wedge (x \geq 0) \wedge (y \geq 0) & : 0 \\ (\neg l) \wedge (x \leq 130) \wedge (y \leq 130) \wedge (x \geq 0) \wedge (y \geq 0) & : -20 \\ else & : -\infty \end{cases}$$

RESERVOIR CONTROL: Reservoir management is a well-studied in OR [?; ?]. In this domain we decided between *drain* (or *not drain*) each reservoir to maximize electricity revenue over the decision-stage horizon while avoiding reservoir overflow and underflow.

We solve a 2-reservoir problem with levels $(l_1, l_2) \in [0, \infty]^2$ with reward penalties for overflow and underflow and a reward gain of 1, i.e.:

$$R = \begin{cases} (l_1 \leq 4500) \wedge (l_2 \leq 4500) \wedge (l_1 \geq 200) \wedge (l_2 \geq 200) & : 1 \\ else & : -\infty \end{cases}$$

The electricity is generated in periods when the $drain()$ action drains water from l_2 to l_1, the other action is $no\text{-}drain()$); we assume a daily control, four days are wet and the next four days are dry (we use three discrete variables to count the day d_1, d_2, d_3), the rainfall replenishment depends on that and is modeled by the noise:

$$n = \begin{cases} (d_1) \wedge (n \leq 2000) \wedge (n \geq 1200) & : legal \\ \neg(d_1) \wedge (n \leq 400) \wedge (n \geq 0) & : legal \\ else & : illegal \end{cases}$$

The transition function for levels of the $drain$ action are

$$l_1' = (n + l_1 - 2800 + 2000)$$
$$l_2' = (n + l_2 - 2000)$$

while for $no\text{-}drain$ action, the *2000* term is dropped.

SPACE TELESCOPE CONTROL: We have extended the problem of slewing a space telescope in order to look a new objective given in [?]. This problem has six actions $a_0, \cdots, a_5$ that change the angle α and the angular rate v. The transition function for a_5 action, when $v < 1\ \frac{deg}{seg}$ and the $z = false$ is:

$$\alpha' = (\alpha + 40.55 * v)$$
$$v' = (2/3v + n)$$
$$z' = (true),$$

Note that we assume a noise in the transition function of the angular rate for a_5, since this action is the only one that changes the zoom of the telescope during the slew. The noise is given by:

$$n = \begin{cases} \neg(z) \wedge (n \leq 0.04 * v) \wedge (n \geq -0.04 * v) & : legal \\ else & : illegal \end{cases}$$

The reward is

[1]While space limitations prevent a self-contained description of all domains, we note that all Java source code and a human/machine readable file format for all domains needed to reproduce the results in this paper can be found online at `http://code.google.com/p/xadd-inference`.

$$R = \begin{cases} (z) \wedge (v \leq 0.02) \wedge (\alpha \leq 1.683) \wedge (v \geq -0.02) \wedge (\alpha \geq 1.283) & : 100 \\ else & : -cost \end{cases}$$

where cost is 0 for action a_0, 1 for actions a_i $i \in \{1,2,3,4\}$ and 10 for action a_5.

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.

[Littman, 1994] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, pages 157–163, 1994.

[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.