An Efficient Algorithm For Discrete-Time Hidden Mode Stochastic Hybrid Systems

Chi-Pang Lam, Allen Y. Yang and S. Shankar Sastry

Abstract— **In this paper, we propose an efficient algorithm to find an optimal control policy in a discrete-time hidden mode stochastic hybrid system, which is a special case of partially observable discrete-time stochastic hybrid systems in which only discrete states are hidden. Many human-centered systems can be modeled as such systems, in which the intent of the human operator is unknown and can be modeled as the hidden mode. In the literature, the optimal control problem of hidden mode stochastic hybrid system is known to have high computational complexity due to the continuous state space. In this paper, we will tackle this computational challenge by using local quadratic functions to approximate the optimal expected reward, which does not have a closed-form expression in general. We will show the efficacy of our proposed method, and the significant improvement in the computational time.**

I. INTRODUCTION

In this paper, we consider a special class in partially observable discrete-time stochastic hybrid system (PODTSHS) [5] in which only discrete states are hidden and there are only discrete control inputs. There are many applications that can be modeled as such systems, especially for human-centered systems in which the intent of the human operator is unknown and can be modeled as the hidden mode. For instance, a driver assistance system should be able to maintain the safety of the driver and the vehicle even though the intent of the driver is unknown [11][12]. For human-robot interaction, it is desirable for a smart robot to infer human intent in order to provide suitable response [3]. More examples could be found in assistive robotics [6][21], multi-agent systems [4], and mobile robot navigation in man-made environments [17].

Hybrid systems with perfect information, in which the states are assumed to be directly observed, have been studied extensively [2][16][8]. But there are only a few works on stochastic hybrid systems with partial information. A general form of discrete-time stochastic hybrid system with partial information can be formulated as a partially observable discrete-time stochastic hybrid system [5][9]. However, the complexity of its computation is still a main issue in solving a general PODTSHS. Instead, one can consider a special case called hidden mode hybrid system, in which only the discrete mode is hidden while the continuous states are assumed to be observed directly [20][19][22]. The safety control problem and mode tracking problem in hidden mode hybrid systems have been studied in [20] and [22] respectively, with the assumption of a deterministic transition map. In the case of hidden mode stochastic hybrid systems, the literature has been focused on the estimation of the hidden mode [7], but not the optimal control policy to control the states, which is the focus of this paper.

In order to find the optimal control policy for a discrete-time hidden mode stochastic hybrid system, we have to maximize the value function at every time step, which is the optimal expected reward over a finite or an infinite horizon. However, it is known that there is no closed-form expression for the value function. Therefore, maximizing the value function at every time step is a challenge. In the past, people either discretize the continuous state space [1] or restrict the probability models and the reward function to be Gaussian [10] in order to approximate the value function as a linear combination of Gaussian functions. Both approaches are either not scalable or too restricted.

Since the model involves hidden states, at every time step, we need to maintain the distribution over the hidden states, known as the belief. Therefore, we are actually doing planning on the belief space rather than the original state space. Researchers have been working on computational techniques for belief space planning [14][18]. In particular, Van den Berg et al. [18] approximated the value functions along the trajectory as quadratic functions. We will adopt the similar technique into our hybrid setting.

In this paper, we use the formulation of PODTSHS and address the optimal control problem in discrete-time hidden mode stochastic hybrid systems with only discrete inputs and cumulative reward. We will show that by using many local quadratic functions to approximate the value function, we can efficiently evaluate the value function at every iteration so the computational time is reduced significantly. In the optimal value function updating process, instead of doing a full update, we only update the lower bound of the optimal value function in order to tackle the integral of a maximization function. Moreover, we draw upon the point-based method for continuous partially observable Markov decision processes (POMDPs) [13] to restrict the number of points of interest used to update the value function. We will show that our method is more efficient and less restricted compared to previous work.

This paper is organized as follows. We first introduce the PODTSHS and derive a general solution to PODTSHS with cumulative reward in Sections II and III. In Section IV, we describe the control problem in discrete-time hidden mode stochastic hybrid systems and propose an algorithm to find the optimal control policy via quadratic approximation.

This work was supported by ONR N000141310341.

Chi-Pang Lam, Allen Y. Yang and S. Shankar Sastry are with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA {cplam,yang,sastry}@eecs.berkeley.edu

Section V shows simulation results, and the conclusion and the future work are in Section VI.

II. Background

A discrete-time stochastic hybrid system was first introduced by Abate et al. [2]. Ding et al. [5] and Lesser [9] extended it to a partially observable framework. We slightly modify the formulation in [5] and [9] and define the partially observable discrete-time stochastic hybrid system as follows:

Definition 1: A partially observable discrete-time stochastic hybrid system (PODTSHS) is a tuple $\mathcal{H} = (\mathcal{Q}, \mathcal{X}, \mathrm{In}, \mathcal{Z}, T_x, T_q, \Omega)$ where

- $\mathcal{Q} = \{q^{(1)}, q^{(2)}, ..., q^{(N_q)}\}$ is a finite set of discrete states.
- $\mathcal{X} \subseteq \mathbb{R}^n$ is a set of continuous states. The hybrid state space is defined by $\mathcal{S} = \mathcal{Q} \times \mathcal{X}$.
- $\mathrm{In} = \Sigma \times \mathcal{U}$, where $\Sigma = \{\sigma^{(1)}, \sigma^{(2)}, ..., \sigma^{(N_\sigma)}\}$ represents a finite set of discrete control inputs affecting the discrete transitions, and $\mathcal{U}$ represents the space of continuous inputs affecting the transition of continuous states.
- $\mathcal{Z} = \mathcal{Z}^q \times \mathcal{Z}^x$ is a nonempty Borel space denoting the observation space, where $\mathcal{Z}^q$ is the observation space of discrete states and $\mathcal{Z}^x$ is the observation space of continuous states.
- $T_x : \mathcal{B}(\mathbb{R}_n) \times \mathcal{Q} \times \mathcal{S} \times \mathrm{In} \to [0,1]$ is a Borel-measurable stochastic kernel which assigns a probability measure to $x_{k+1} \in \mathcal{X}$ given $s_k \in \mathcal{S}, \sigma_k \in \Sigma, u_k \in \mathcal{U}$ and $q_{k+1} \in \mathcal{Q}$: $T_x(dx_{k+1}|q_{k+1}, s_k, \sigma_k, u_k)$.
- $T_q : \mathcal{Q} \times \mathcal{X} \times \mathrm{In} \to [0,1]$ is a discrete transition kernel assigning a probability distribution to $q_{k+1} \in \mathcal{Q}$ given $s_k \in \mathcal{S}, \sigma_k \in \Sigma$ and $u_k \in \mathcal{U} : T_q(q_{k+1}|s_k, \sigma_k, u_k)$.
- $\Omega : \mathcal{B}(\mathcal{Z}) \times \mathcal{S} \times \mathrm{In} \to [0,1]$ is a Borel-measurable stochastic kernel assigning a probability measure to $z_k \in \mathcal{Z}$ given $s_k \in \mathcal{S}, u_{k-1} \in \mathcal{U}$ and $\sigma_{k-1} \in \Sigma$: $\Omega(dz_k|s_k, \sigma_{k-1}, u_{k-1})$.

To simplify the problem we make the following assumptions:

1) The discrete transition T_q only depends on $q_k \in \mathcal{Q}$ and $\sigma_k \in \Sigma$: $T_q(q_{k+1}|s_k, \sigma_k, u_k) = T_q(q_{k+1}|q_k, \sigma_k)$.
2) The continuous transition T_x only depends on $q_{k+1} \in \mathcal{Q}$, $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$: $T_x(dx_{k+1}|q_{k+1}, s_k, \sigma_k, u_k) = T_x(dx_{k+1}|q_{k+1}, x_k, u_k)$.
3) The measurement kernel Ω does not depend on the inputs and can be factorized into measurements for discrete states and measurements for continuous states: $\Omega(dz_k|s_k, \sigma_{k-1}, u_{k-1}) = \Omega_q(z^q|q_k) \times \Omega_x(dz^x|x_k)$.

Here we use a driver assistance example to illustrate the relationship between the general PODTSHS and the above simplification. We assume the driver could be drowsy or awake, which is modeled as the hidden discrete mode q. The continuous state x is the position of the car. The discrete input σ indicates whether the warning signal is turned on to awake the driver, and the continuous input u is an augmented control input to the car. The first assumption means whether the driver is drowsy depends on whether she is drowsy at the previous state and whether the warning signal is turned on to awake her. The second assumption means that the position of the vehicle depends on whether the human is awake, the previous position of the vehicle and the augmented control input. The last assumption means we measure the state of the human and the state of the car separately. A more concrete example of human-in-the-loop system can be found in Section V.

Under this PODTSHS model, the information up to step k is denoted as $i_k = (\sigma_0, u_0, z_1, \sigma_1, u_1, z_2, \ldots, \sigma_{k-1}, u_{k-1}, z_k)$, along with the prior distribution of the initial state s_0. Working directly with the information state is cumbersome, so instead we work with the distribution of states at every time step, which is known as belief state. The belief state is defined as follows:

Definition 2: A belief $b(s)$ is a probability distribution over $\mathcal{S}$ with $\int_{s \in \mathcal{S}} b(s) \mathrm{d}s = 1$. Since $s = (q, x)$ is a hybrid state, the integral over $s \in \mathcal{S}$ is defined as $\int_{s \in \mathcal{S}} f(s) \mathrm{d}s = \sum_{q \in \mathcal{Q}} \int_{x \in \mathcal{X}} f(q, x) \mathrm{d}x$.

The belief changes every time step. We denote the new belief at time $k+1$ when executing control inputs (σ_k, u_k) and observing new measurement z_{k+1} as $b_{k+1}^{\sigma_k, u_k, z_{k+1}}(s_{k+1})$. The belief can be updated recursively by:

$$\begin{aligned} & b_{k+1}^{\sigma_k, u_k, z_{k+1}}(s_{k+1}) = P(s_{k+1}|\sigma_k, u_k, z_{k+1}, b_k) \\ =& \frac{P(z_{k+1}|s_{k+1}, \sigma_k, u_k, b_k) P(s_{k+1}|\sigma_k, u_k, b_k)}{P(z_{k+1}|\sigma_k, u_k, b_k)} \\ =& \eta \Omega(z_{k+1}|s_{k+1}, \sigma_k, u_k) \times \\ & \int_{s_k \in \mathcal{S}} T_x(dx_{k+1}|q_{k+1}, x_k, u_k) T_q(q_{k+1}|q_k, \sigma_k) b_k(s_k) \mathrm{d}s_k, \end{aligned} \tag{1}$$

where η is a normalization factor.

Definition 3: A policy π for $\mathcal{H}$ is a sequence $\pi = (\pi_0, \pi_1, \pi_2, \cdots)$, where $\pi_k(b_k) \in \Sigma \times \mathcal{U}$ is a map from the belief state at time k to the set of controls.

A reward function is denoted as $R(q, x, \sigma, u)$ or $R_{\sigma,u}(q, x) \in \mathbb{R}$, which is obtained by the system if it executes (σ, u) when the system is in state (q, x). To assess the quality of a given policy π, we use a value function to represent the expected m-step cumulative reward starting from the belief state b_0:

$$J_m^\pi(b_0) = \mathbb{E}[\sum_{k=0}^{m} \gamma^k R(s_k, \sigma_k, u_k)], \tag{2}$$

where $0 \le \gamma \le 1$ is a discount factor and the controls $(\sigma_k, u_k) = \pi_k(b_k)$. The optimal value function at step m is $J_m^* = \max_\pi J_m^\pi$. For all $m = 0, 1, 2, \cdots$, the optimal value function can be calculated by

$$J_{m+1}^*(b) = \max_{(\sigma,u) \in \Sigma \times \mathcal{U}} \{\langle R_{\sigma,u}, b \rangle + \gamma \int_z p(z|\sigma, u, b) J_m^*(b^{\sigma,u,z}) \mathrm{d}z\}, \tag{3}$$

where the operator $\langle \cdot, \cdot \rangle$ is defined as $\langle f(q,x), g(q,x) \rangle = \sum_{q \in Q} \int_{x \in \mathcal{X}} f(q,x) g(q,x) \mathrm{d}x$.

The goal of a PODTSHS with cumulative reward is to find an optimal policy to maximize the m-step value function to yield $J_m^* = \max_\pi J_m^\pi$. For infinite horizon, i.e., $m \to \infty$, the optimal policies of all time steps are the same, i.e., $\pi^* = \pi_0 = \pi_1 = \cdots$.

By Lemma 1 in [13], we know that the m-step optimal value function can be expressed as:

$$J^*_m(b) = \max_{\{\alpha^i_m\}_i} \langle \alpha^i_m, b\rangle, \tag{4}$$

for an appropriate continuous set of α-functions $\alpha^i_m : \mathcal{S} \to \mathbb{R}$. Therefore, to find the optimal value function, it is equivalent to find the set of α-functions $\{\alpha^j_m\}_j$.

III. Recursive Update of the Value Function

Using the α-function formulation, we will derive a recursive update process for the set of α-functions. For $m = 1$, the optimal value function is the maximum of the instant reward:

$$J^*_1(b) = \max_{(\sigma,u)} \langle R_{(\sigma,u)}, b\rangle. \tag{5}$$

By Comparing (5) to (4), we can see that the first step α-functions $\{\alpha^j_1\}_j$ are $\{R_{(\sigma,u)}\}_{(\sigma,u)}$. The $(m+1)$-step α-functions $\{\alpha^j_{m+1}\}_j$ can be calculated from the m-step α-functions $\{\alpha^j_m\}_j$. Starting from (3), we have:

$$J^*_{m+1}(b) = \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b\rangle + \gamma \int_z p(z|\sigma,u,b) J^*_m(b^{\sigma,u,z}) \mathrm{d}z \right\} \tag{6}$$

$$= \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b\rangle + \gamma \int_z p(z|\sigma,u,b) \max_{\{\alpha^j_m\}_j} \langle \alpha^j_m, b^{\sigma,u,z}\rangle \mathrm{d}z \right\}$$

$$= \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b\rangle + \gamma \int_z \max_{\{\alpha^j_m\}_j} \int_s b(s) \int_{s'} \alpha^j_m(s') \right.$$

$$\left. \Omega(z|s',\sigma,u) T_x(x'|q',x,u) T_q(q'|q,\sigma) \mathrm{d}s' \mathrm{d}s \mathrm{d}z \right\}.$$

Let

$$\alpha^j_{\sigma,u,z}(s) = \int_{s'} \alpha^j_m(s') \Omega(z|s',\sigma,u) T_x(x'|q',x,u) T_q(q'|q,\sigma) \mathrm{d}s', \tag{7}$$

then we have:

$$J^*_{m+1}(b) = \max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b\rangle + \gamma \int_z \max_{\{\alpha^j_m\}_j} \langle \alpha^j_{\sigma,u,z}, b\rangle \mathrm{d}z \right\}. \tag{8}$$

Let

$$(\sigma^*, u^*) = \arg\max_{(\sigma,u)\in\Sigma\times\mathcal{U}} \left\{ \langle R_{\sigma,u}, b\rangle + \gamma \int_z \max_{\{\alpha^j_m\}_j} \langle \alpha^j_{\sigma,u,z}, b\rangle \mathrm{d}z \right\}. \tag{9}$$

Then if we represent $J^*_{m+1}(b)$ as the form of inner product as in (4), we can find that a new $(m+1)$-step α-function for a specific belief b can be written as:

$$\alpha^b_{(\sigma^*,u^*)}(s) = R_{\sigma^*,u^*}(s) + \gamma \sum_{z^q\in\mathcal{Z}^q} \int_{z^x} \arg\max_{\{\alpha^j_{\sigma^*,u^*,z}\}_j} \langle \alpha^j_{\sigma^*,u^*,z}, b\rangle \mathrm{d}z^x. \tag{10}$$

Then the new set of α-functions is:

$$\{\alpha^i_{m+1}\}_i = \bigcup_{\forall b} \{\alpha^b_{\sigma^*,u^*}\}. \tag{11}$$

Given the set of α-functions, and a belief b, the policy function $\pi(\cdot)$ is the map from b to the optimal control calculated by (9).

Although we derive the updating process of the set of α-functions theoretically, it is very challenging to perform the exact update in practice. There are four reasons:

1) We have to maximize the non-convex value function over continuous input space;
2) There is no efficient way to find the exact value of the integral of maximization function in (10);
3) There is no closed-form expression for α-functions;
4) It is not possible to find the full set of α-functions for all b in the belief space because the belief space is of infinite dimension with continuous state variables.

IV. Approximate Solution to a Discrete Time Hidden Mode Stochastic Hybrid System

Instead of dealing with the general PODTSHS, we consider a special case of PODTSHS where only discrete states are hidden and there are only discrete inputs. In this case we will avoid the first challenge about the continuous input space. Although we do not consider continuous inputs, we will show in the simulation in Section V that we can use a controller selection scheme to introduce continuous control inputs in the system. More specifically, we consider a PODTSHS as follows:

1) $\mathcal{U} = \emptyset$;
2) $\mathcal{Z}^x = \mathcal{X}$;
3) $\Omega_x(z^x|x_k) = \delta(z^x - x_k)$.

We also model the dynamical system under each discrete mode q as

$$x_{k+1} = f_q(x_k) + w, \quad w \sim \mathcal{N}(0, W_q),$$

where w is the Gaussian noise and W_q is the covariance matrix of w at discrete mode q. We also assume that $f_q(x)$ is differentiable. The above dynamical system implies that the continuous transition $T_x(x_{k+1}|q_{k+1}, x_k)$ is a Gaussian function with mean $f_{q_{k+1}}(x_k)$ and covariance $W_{q_{k+1}}$, i.e. $\mathcal{N}(f_{q_{k+1}}(x_k), W_{q_{k+1}})$.

In this case, since the continuous states are observable, the belief at any time step k will have the following form:

$$b_k(q_k, x_k) = \begin{cases} b_k(q_k, z_k) \geq 0, & \text{if} \quad x_k = z_k; \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The belief update (1) becomes:

$$\begin{aligned} & b^{\sigma_k, z_{k+1}}_{k+1}(q_{k+1}, x_{k+1}) \\ =& \eta \Omega(z^q_{k+1}|q_{k+1}) \delta(z^x_{k+1} - x_{k+1}) \times \\ & \sum_{q_k\in\mathcal{Q}} \int_{x_k\in\mathcal{X}} T_x(x_{k+1}|q_{k+1}, x_k) T_q(q_{k+1}|q_k, \sigma_k) b_k(q_k, x_k) \mathrm{d}x_k \\ =& \begin{cases} \eta \Omega(z^q_{k+1}|q_{k+1}) T_x(z_{k+1}|q_{k+1}, z_k) \times & \\ \sum_{q_k\in\mathcal{Q}} T_q(q_{k+1}|q_k, \sigma_k) b_k(q_k, z_k), & \text{if} \quad x_{k+1} = z_{k+1}; \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{13}$$

where

$$\eta = \sum_{q_{k+1}} \Omega(z^q_{k+1}|q_{k+1}) T_x(z_{k+1}|q_{k+1},z_k) T_q(q_{k+1}|q_k,\sigma_k) b_k(q_k,z_k).$$

We can get the optimal value J^*_{m+1} with (8):

$$J^*_{m+1}(b) = \max_{\sigma\in\Sigma} \left\{ \langle R_\sigma, b\rangle + \gamma \sum_{z^q\in\mathcal{Z}^q} \int_{z^x} \max_{\{\alpha^j_m\}_j} \langle \alpha^j_{\sigma,z^q,z^x}, b\rangle \mathrm{d}z^x \right\}, \tag{14}$$

where by equation (7), $\alpha^j_{\sigma,z^q,z^x}(q,x)$ is:

$$\alpha^j_{\sigma,z^q,z^x}(q,x) = \sum_{q'\in\mathcal{Q}} \alpha^j_m(q',z^x)\Omega(z^q|q')T_x(z^x|q',x)T_q(q'|q,\sigma). \tag{15}$$

A. Quadratic Approximation for α-Functions

In order to evaluate the optimal value $J^*_{m+1}(\bar{b})$ for a specific belief $\bar{b}$, (by (12), without loss of generality, assume $\bar{b}(q,x) \geq 0$ only if $x = \bar{x}$), we have to deal with the integral of a maximization function in (14). However, as we mentioned before, there is no efficient way to calculate an exact closed-form solution of the integral of a maximization function. To tackle this challenge, instead of directly calculate the integral, we calculate a lower bound of the optimal value $J^*_{m+1}(\bar{b})$ by the inequality:

$$\int_{z^x} \max_{\{\alpha^j_m\}_j} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x \geq \max_{\{\alpha^j_m\}_j} \int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x. \tag{16}$$

Using the lower bound is important because it will not overestimate the optimal value function. Overestimation may lead to divergence of J_{m+1} because we find J_{m+1} in a maximization scheme. We also propose to use a quadratic function to approximate the α-function in order to tackle the third challenge, i.e., let $\alpha^j(q,x) \approx a^j_0(q) + a^j_1(q)^T x + x^T A^j_2(q) x$. We will show that by doing so, we can calculate a closed-form lower bound of the optimal value $J^*_{m+1}(\bar{b})$. The integration in (16) can be obtained by:

$$\begin{aligned}
&\int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x \\
&= \int_{z^x} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \alpha^j_m(q',z^x)\Omega(z^q|q')T_x(z^x|q',\bar{x})T_q(q'|q,\sigma)\bar{b}(q,\bar{x})\mathrm{d}z^x \\
&= \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\bar{x}) \int_{z^x} \alpha^j_m(q',z^x)T_x(z^x|q',\bar{x})\mathrm{d}z^x \\
&= \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\bar{x})\mathbb{E}[\alpha^j_m(q',z^x)],
\end{aligned} \tag{17}$$

where

$$\mathbb{E}[\alpha^j_m(q',z^x)] = a^j_0(q') + (a_1(q')^j)^T \mathbb{E}[z^x] + \mathbb{E}[(z^x)^T A^j_2(q) z^x].$$

Since $T_x(z^x|q',\bar{x})$ is a Gaussian distribution with mean $f_{q'}(\bar{x})$ and covariance $W_{q'}$, we have:

$$\begin{aligned}
\mathbb{E}[\alpha^j_m(q',z^x)] = a^j_0(q') + (a^j_1(q'))^T f_{q'}(\bar{x}) + \\
(f_{q'}(\bar{x}))^T A^j_2(q') f_{q'}(\bar{x}) + tr(A^j_2(q')W_{q'}).
\end{aligned} \tag{18}$$

In (18), we are using the fact that $\mathbb{E}[x^T L x] = \mathbb{E}[x]^T L \mathbb{E}[x] + \mathrm{Tr}(L\,\mathrm{Var}(x))$. Combining (14), (16), (17) and (18), we can get a lower bound of $J^*_{m+1}(\bar{b})$. Let

$$\alpha^*_m = \arg\max_{\{\alpha^j_m\}_j} \int_{z^x} \langle \alpha^j_{\sigma,z^q,z^x}, \bar{b}\rangle \mathrm{d}z^x, \tag{19}$$

then the lower bound of J^*_{m+1} is

$$\begin{aligned}
&J^*_{m+1}(\bar{b}) \geq \max_{\sigma\in\Sigma} \Bigg\{ \langle R_\sigma, \bar{b}\rangle + \\
&\gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\bar{x})\mathbb{E}[\alpha^*_m(q',z^x)] \Bigg\}.
\end{aligned}$$

Let

$$\begin{aligned}
\sigma^* = &\arg\max_{\sigma\in\Sigma} \Bigg\{ \langle R_\sigma, \bar{b}\rangle + \\
&\gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q\in\mathcal{Q}} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma)\bar{b}(q,\bar{x})\mathbb{E}[\alpha^*_m(q',z^x)] \Bigg\}.
\end{aligned} \tag{20}$$

Then similar to (10), a new α_{m+1} can be updated by:

$$\begin{aligned}
&\alpha_{m+1}(q,x) = R_{\sigma^*}(q,x) + \\
&\quad \gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q'\in\mathcal{Q}} \Omega(z^q|q')T_q(q'|q,\sigma^*)\mathbb{E}[\alpha^*_m(q',z^x)].
\end{aligned} \tag{21}$$

To maintain the quadratic form of the α-function, we approximate $\alpha_{m+1}(q,x)$ as a quadratic function around $\bar{x}$:

$$\begin{aligned}
\alpha_{m+1}(q,x) \approx &\alpha_{m+1}(q,\bar{x}) + \left(\frac{\partial \alpha_{m+1}(q,x)}{\partial x}\bigg|_{\bar{x}} \right)^T (x-\bar{x}) + \\
&\frac{1}{2}(x-\bar{x})^T \frac{\partial^2 \alpha_{m+1}(q,x)}{\partial x \partial x}\bigg|_{\bar{x}} (x-\bar{x}).
\end{aligned} \tag{22}$$

We also linearize the dynamical system around $\bar{x}$:

$$x_{k+1} - f_q(\bar{x}) = H_q(x_k - \bar{x}), \tag{23}$$

where $H_q = Df_q(x)\big|_{\bar{x}}$. Let the quadratic approximation of $R_{\sigma^*}(q,x)$ around $\bar{x}$ be

$$R_{\sigma^*}(q,x) \approx R_{\sigma^*}(q,\bar{x}) + r_1^T(x-\bar{x}) + \frac{1}{2}(x-\bar{x})^T M (x-\bar{x}), \tag{24}$$

where $r_1 = \frac{\partial R_{\sigma^*}(q,x)}{\partial x}\big|_{\bar{x}}$ and $M = \frac{\partial^2 R_{\sigma^*}(q,x)}{\partial x \partial x}\big|_{\bar{x}}$. Combining (21), (23) and (24) we can get:

$$\begin{aligned}
\frac{\partial \alpha_{m+1}(q,x)}{\partial x}\bigg|_{\bar{x}} = &r_1 + \gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q'\in\mathcal{Q}} \Bigg[\Omega(z^q|q')T_q(q'|q,\sigma^*) \times \\
&\left(H^T_{q'} a^*_1(q') + 2H^T_{q'} A^*_2(q') f_{q'}(\bar{x}) \right) \Bigg]
\end{aligned} \tag{25}$$

and

$$\begin{aligned}
\frac{\partial^2 \alpha_{m+1}(q,x)}{\partial x \partial x}\bigg|_{\bar{x}} = &M + \gamma \sum_{z^q\in\mathcal{Z}^q} \sum_{q'\in\mathcal{Q}} \Bigg(\Omega(z^q|q')T_q(q'|q,\sigma^*) \times \\
&2H^T_{q'} A^*_2(q') H_{q'} \Bigg).
\end{aligned} \tag{26}$$

To summarize, we can update a new α-function for a specific belief $\bar{b}$ by Algorithm 1. Since for every α-function, there is a specific linearizing point x used for quadratic approximation, we are not using the whole set of α_m^j's, but using those whose linearizing points are closed enough to $\bar{x}$ to perform update in Step 1 of Algorithm 1.

Algorithm 1: α-function update

```
Function Update($\{a_m^j\}_j$, $\bar{b}$)
    1. Obtain $\alpha_m^*$ by (19) where $\int_{z^x}\langle\alpha_{\sigma,z^q,z^x}^j,\bar{b}\rangle \mathrm{d}z^x$
       can be calculated by (17) and (18).
    2. Get $\sigma^*$ by (20) and (18).
    3. Obtain the quadratic approximation of a new
       $\alpha$-function $\alpha_{m+1}$ by (22), (25) and (26).
    return $\alpha_{m+1}$
```

A full updating process requires updating $\{\alpha_{m+1}^j\}_j$ over all $\bar{b} \in \mathcal{B}$, the entire belief space. However, as we mentioned before, the belief of continuous states is of infinite dimension, so finding the full set of the $(m+1)$-step α-functions is not possible. The point-based method for POMDP suggests only using a finite number of reachable beliefs to update α-functions and also bounding the number of new α-functions. The point-based method allows us to update α-functions in bounded times, which makes the problem tractable. Therefore, we will adopt the point-based method to tackle this challenge.

There are different variations of point based method in which people use different methods for generating belief set B and updating a new set of α-functions. We propose Algorithm 2 to perform point based value iteration for hidden mode stochastic hybrid system.

The first step of Algorithm 2 is to generate a set of reachable beliefs. We first randomly explore the belief space and then use K-means to cluster the belief set. After that, we select beliefs from each cluster randomly until it meets the predefined number of beliefs. Since we found that random exploration in PODTSHS will result in many similar beliefs, clustering them and selecting them from different clusters can increase the diversity of beliefs, which accelerates the value iteration process in next step. We adopt Perseus algorithm [15] to perform point-based value iteration which has been shown to be efficient for discrete POMDP. In every iteration of `ValueIteration`, the time complexity is $\mathcal{O}(N_B|\Sigma||\mathcal{Z}^q||\mathcal{Q}|^2|V_\alpha|n^2)$, where N_B is the number of beliefs used for update, $|\Sigma|$ is the number of discrete inputs, $|\mathcal{Z}^q|$ is the number of discrete observations, $|\mathcal{Q}|$ is the number of discrete states, $|V_\alpha|$ is the number of α-functions at every iteration, and n is the dimension of the continuous state.

Algorithm 2: Value iteration for discrete-time hidden mode stochastic hybrid system

```
Input: Hidden mode stochastic hybrid system $\mathcal{H}$, initial
       state $(q_0, x_0)$ and the number of beliefs $N_B$
Output: $V_\alpha$: The set of $\alpha$-functions

$B$ = BeliefCollection($(q_0,x_0)$,$N_B$)
$V_\alpha$ = ValueIteration ($B$)
Function BeliefCollection($(q,x)$,$N_B$)
    repeat
        Uniformly choose $\sigma$ from $\Sigma$
        Sample $(q',x')' \sim T_x(x'|q',x)T_q(q'|q,\sigma)$
        Sample $z^q \sim \Omega(z^q|q')$
        $b' = b^{\sigma,z}$ by (13)
        $B \leftarrow B \bigcup b'$
        $(q,x) \leftarrow (q',x')$
    until $|B| = 10N_B$;
    Clustering $B$ by K-means: $C =$K-means($B$)
    $B' \leftarrow \emptyset$
    repeat
        Randomly select a cluster $C_i$ and randomly
        select a belief $b$ from $C_i$
        $B' \leftarrow B' \bigcup b$, $C_i \leftarrow C_i \setminus b$
    until $|B'| = N_B$;
    return $B'$

Function ValueIteration($B$)
    $V_\alpha \leftarrow \{R_\sigma\}_{\sigma\in\Sigma}$
    repeat
        $B' \leftarrow B$; $V'_\alpha \leftarrow \emptyset$
        while $B' \neq \emptyset$ do
            Choose $\bar{b} \in B'$ randomly
            $\alpha' \leftarrow$ Update$(V_\alpha,\bar{b})$ by Algorithm 1
            if $\langle\alpha',\bar{b}\rangle \geq J^*(\bar{b})$ ($J^*(\bar{b})$ is calculated by
            (4)) then
                $B' \leftarrow \{b \in B' | \langle\alpha',b\rangle < J^*(b)\}$
                $\alpha_b \leftarrow \alpha'$
            else
                $B' \leftarrow B' \setminus b$
                $\alpha_b \leftarrow \arg\max_{\alpha\in V_\alpha}\langle\alpha,b\rangle$
            $V'_\alpha \leftarrow V'_\alpha \bigcup \alpha_b$
        $V_\alpha \leftarrow V'_\alpha$
    until $\forall b \in B$, $V_\alpha(b)$ converges;
    return $V_\alpha$
```

V. Simulation Results

We use two simulations to demonstrate the efficacy and the speed of the proposed method. The simulations are programmed in C++ on a laptop running Mac OS X with 2GHz Quad-core Intel Core i7. In the first simulation, we simulate a human-in-the-loop system. It shows that although we only consider discrete inputs in our proposed algorithm, we can actually use a controller selection scheme to introduce continuous inputs. The second simulation compared our method with a discretization scheme [1]. To our best knowledge, we are aware of another computational method in [10], which uses the linear combination of Gaussian functions to approximate the α-functions. However, it requires the probability models and the reward function to be Gaussian, which is not applicable to our case.

The first simulation models a human-in-the-loop system with a two-dimensional continuous state space, in which a driver, who could be either attentive or distracted, is keeping the car at the middle of a lane. x is the position and v

is the velocity of the car vertical to the direction of the lane. Suppose that there are two feedback systems. One is a warning system that reminds the driver to be attentive, and the other one is an augmented control input u_m obtained by controllers C_0 that will not intervene the driver, or C_1 that will help driving the car toward the middle of the lane. In such setting, we use a controller selection scheme to introduce continuous input u_m.

More specifically, the hidden mode stochastic hybrid system is defined as follows:

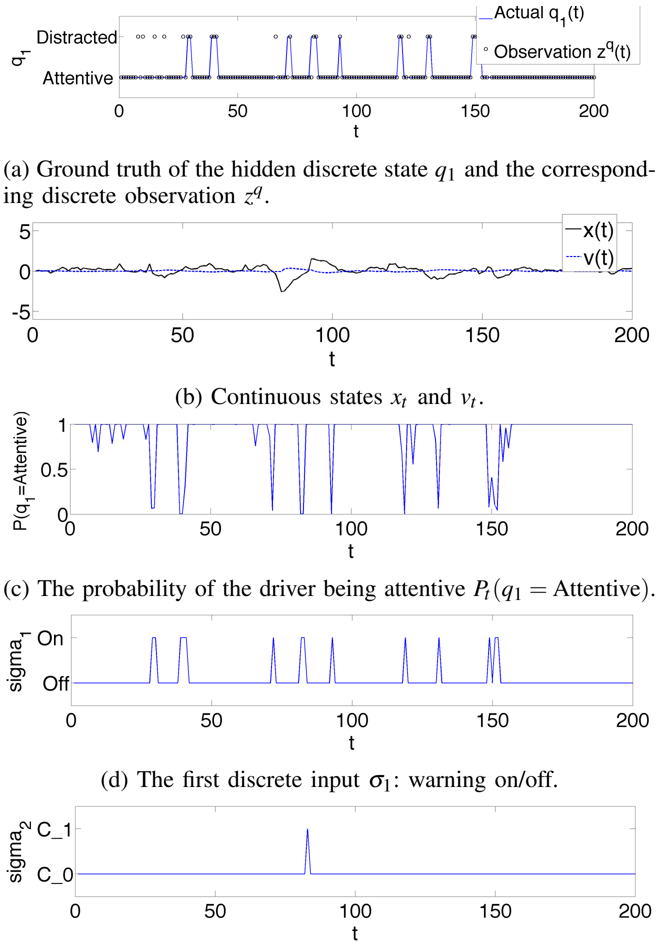
- $\mathcal{Q}_1$=$\{q^a$=Attentive, q^d = Distracted$\}$, $\mathcal{Q}_2 = \{q^{(0)} = C_0, q^{(1)} = C_1\}$. Hidden state space $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$.
- Continuous state $[x,v]^T \in \mathbb{R}^2$.
- $\Sigma_1 = \{\sigma^w$ = Warning, σ^{nw} = No warning$\}$ and $\Sigma_2 = \{\sigma^{(0)}$ = Execute C_0, $\sigma^{(1)}$ = Execute $C_1\}$. The set of discrete controls is $\Sigma = \Sigma_1 \times \Sigma_2$
- $\mathcal{Z}^q = \mathcal{Q}_1$.
- $T_q(q'|q,\sigma) = T_{q_1}(q_1|q_1,\sigma_1)T_{q_2}(q_2'|q_2,\sigma_2)$ where $T_{q_1}(q_1' = q_1|q_1,\sigma_1 = \sigma^{nw}) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^a,\sigma_1) = 0.95$, $T_{q_1}(q_1' = q^a|q_1 = q^d,\sigma_1 = \sigma^w) = 0.8$ and $T_{q_2}(q_2' = \sigma_2|q_2,\sigma_2) = 1$.
- $$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{bmatrix} u_h + \begin{bmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{bmatrix} u_m + w. \tag{27}$$

  $$u_{h,k} = \begin{cases} -[K_1 \quad K_2] \begin{bmatrix} x_k \\ v_k \end{bmatrix}, & \text{if } q_1 = q^a; \\ 0, & \text{if } q_1 = q^d. \end{cases}$$

  $$u_{m,k} = \begin{cases} 0, & \text{if } q_2 = q^{(0)}; \\ -[K_1 \quad K_2] \begin{bmatrix} x_k \\ v_k \end{bmatrix}, & \text{if } q_2 = q^{(1)}, \end{cases}$$

  where K_1 and K_2 are feedback gains such that the system is stable, and $w \sim \mathcal{N}(0, \left(\begin{smallmatrix} 0.2 & 0 \\ 0 & 0 \end{smallmatrix}\right))$ when $q_1 = q^a$ and $w \sim \mathcal{N}(0, \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right))$ when $q_1 = q^d$.
- $\Omega(z^q = q_1|q_1) = 0.95$.
- $R(q,x,v,\sigma) = 100 - [x \quad v] \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} - 5\mathbb{I}(\sigma_1 = \sigma^w) - 5\mathbb{I}(\sigma_2 = \sigma^{(1)})$, where $\mathbb{I}(\cdot)$ is the identity function.

Given the discrete-time hidden mode stochastic hybrid system, we first compute the optimal control policy $\pi(\cdot)$ by Algorithm 2, which takes 27s with 5000 belief states. We then evaluate our policy by the following simulation process: based on the current belief b_t, we obtain the control $\sigma_t = \pi(b_t)$, apply σ_t to the system and sample a new discrete state q_{t+1} from T_q. We calculate x_{t+1} by equation (27) and sample new observation z^q_{t+1} from Ω, by which we update a new belief b_{t+1} and the whole process repeats. Figure 1a shows the ground truth of the hidden discrete state q_1 and figure 1b shows the continuous state x. Figure 1c shows the marginal belief $P_t(q_1)$ of every time step, and figures 1d and 1e show the corresponding controls obtained by our learned policy.



(a) Ground truth of the hidden discrete state q_1 and the corresponding discrete observation z^q.

(b) Continuous states x_t and v_t.

(c) The probability of the driver being attentive $P_t(q_1 = \text{Attentive})$.

(d) The first discrete input σ_1: warning on/off.

(e) The second discrete input σ_2: the selected controller.

Fig. 1: Simulation results for a human-in-the-loop system.

Intuitively, the goal of the policy should encourage the system staying in mode q^a and keeping x and v zero. The simulation result conforms with our intuition that when $P(q_1 = \text{Attentive})$ goes down to some threshold, there will be a warning σ_1 = Warning in order to keep the driver attentive. Moreover, we can see that our learned policy selects controller C_1 when the x is too far from zero. This simulation shows that using quadratic approximation, we can still get a reasonable control policy.

Finally, we compare the time used to get the policy in our proposed algorithm with a discretization scheme [1]. In this simulation, we reduce the above 2D example to a 1D example with only one continuous variable x. Table I shows the computing time, in which we can see that our proposed algorithm is at least 130 times faster than the discretization scheme. Moreover, we compare the average reward by running 50 simulations for both schemes. As shown in Table II, our method gets a higher average reward than the discretization scheme. Hence, out method outperforms the discretization scheme. The main reason is that the accuracy of discretization highly depends on how fine you discretize the state space. If you do not discretize the space fine enough, the error will be large, but if we discretize it too fine, the computation becomes slower. We can see that our method both increase the efficiency and retain the optimality of the policy.

TABLE I: The computational time of our method and the traditional discretization scheme.

	Number of belief $\|\mathcal{B}\|$ used in updating value function				
	100	500	1000	2500	5000
Discretization	71m	93m	114m	120m	132m
Our method	1.0s	5.7s	12.4s	34.7s	61s

TABLE II: The average reward of our method and the traditional discretization scheme.

	Number of belief $\|\mathcal{B}\|$ used in updating value function				
	100	500	1000	2500	5000
Discretization	9912	9914	9917	9918	9915
Our method	9917	9918	9919.2	9919.6	9920

VI. CONCLUSION AND FUTURE WORK

We have proposed an algorithm to find an approximate optimal control policy for the hidden model stochastic hybrid system. We have shown that by approximating α-functions as quadratic functions and using lower bound of the optimal value function to do update, we can efficiently perform value iteration in order to find the optimal control policy. We have compared our method with the traditional discretization scheme and have shown that our method can find the optimal policy faster while still remain the optimality of the control policy.

For future work, we would like to find a theoretic guarantee on the bound of the optimal value function using our method and to explore how to generalize this technique to a general PODTSHS.

REFERENCES

[1] Alessandro Abate, Saurabh Amin, Maria Prandini, John Lygeros, and Shankar Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo, editors, *Hybrid Systems: Computation and Control*, volume 4416 of *Lecture Notes in Computer Science*, pages 4–17. Springer Berlin Heidelberg, 2007.

[2] Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724 – 2734, 2008.

[3] D Kulić EA Croft. Estimating intent for human-robot interaction. In *IEEE International Conference on Advanced Robotics*, pages 810–815, 2003.

[4] Yiannis Demiris. Prediction of intent in robotics and multi-agent systems. *Cognitive Processing*, 8(3):151–158, 2007.

[5] J. Ding, A. Abate, and C. Tomlin. Optimal control of partially observable discrete time stochastic hybrid systems for safety specifications. In *American Control Conference (ACC), 2013*, pages 6231–6236, June 2013.

[6] M.S. Erden and T. Tomiyama. Human-intent detection and physically interactive control of a robot without force sensors. *Robotics, IEEE Transactions on*, 26(2):370–382, April 2010.

[7] Michael W. Hofbaur and Brian C. Williams. Mode estimation of probabilistic hybrid systems. In Claire J. Tomlin and Mark R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, volume 2289 of *Lecture Notes in Computer Science*, pages 253–266. Springer Berlin Heidelberg, 2002.

[8] M. Kamgarpour, J. Ding, S. Summers, A. Abate, J. Lygeros, and C. Tomlin. Discrete time stochastic hybrid dynamical games: Verification amp; controller synthesis. In *Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on*, pages 6122–6127, Dec 2011.

[9] K. Lesser and M. Oishi. Reachability for partially observable discrete time stochastic hybrid systems. *Automatica*, 2014.

[10] Kendra Lesser and Meeko Oishi. Computational techniques for reachability analysis of partially observable discrete time stochastic hybrid systems. Technical Report arXiv:1404.5906, 2014.

[11] A. Liu and A. Pentland. Towards real-time recognition of driver intentions. In *Intelligent Transportation System, IEEE Conference on*, pages 236–241, Nov 1997.

[12] Alex Pentland and Andrew Lin. Modeling and prediction of human behavior. *Neural Computation*, 11:229–242, 1995.

[13] Josep M. Porta, Nikos Vlassis, Matthijs T.J. Spaan, and Pascal Poupart. Point-based value iteration for continuous pomdps. *Journal of Machine Learning Research*, 7:2329–2367, December 2006.

[14] Samuel Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 2009.

[15] Matthijs T. J. Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for pomdps. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[16] Sean Summers and John Lygeros. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica*, 46(12):1951 – 1961, 2010.

[17] S. Thompson, T. Horiuchi, and S. Kagami. A probabilistic model of human motion and navigation intent for mobile robot path planning. In *Autonomous Robots and Agents, 4th International Conference on*, pages 663–668, Feb 2009.

[18] Jur van den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

[19] R. Verma and D. Del Vecchio. Control of hybrid automata with hidden modes: Translation to a perfect state information problem. In *Decision and Control, 49th IEEE Conference on*, pages 5768–5774, Dec 2010.

[20] R. Verma and D. Del Vecchio. Safety control of hidden mode hybrid systems. *Automatic Control, IEEE Transactions on*, 57(1):62–77, Jan 2012.

[21] G. Wasson, P. Sheth, M. Alwan, K. Granata, A. Ledoux, and Cunjun Huang. User intent in a shared control framework for pedestrian mobility aids. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, volume 3, pages 2962–2967 vol.3, Oct 2003.

[22] Sze Zheng Yong and E. Frazzoli. Hidden mode tracking control for a class of hybrid systems. In *American Control Conference (ACC)*, pages 5735–5741, June 2013.