Abstract—Human–robot collaboration (HRC) has emerged as a hot research area at the intersection of control, robotics, and psychology in recent years. It is of critical importance to obtain an expressive but meanwhile tractable model for human beings in HRC. In this letter, we propose a model called vector autoregressive partially observable Markov decision process (VAR-POMDP) which is an extension of the traditional POMDP model by considering the correlation among observations. The VAR-POMDP model is more powerful in the expressiveness of features than the traditional continuous observation POMDP since the traditional one is a special case of the VAR-POMDP model. Meanwhile, the proposed VAR-POMDP model is also tractable, as we show that it can be effectively learned from data and we can extend point-based value iteration (PBVI) to VAR-POMDP planning. Particularly, in this letter, we propose to use the Bayesian non-parametric learning to decide potential human states and learn a VAR-POMDP model using data collected from human demonstrations. Then, we consider planning with respect to PCTL which is widely used as safety and reachability requirement in robotics. Finally, the advantage of using the proposed model for HRC is validated by experimental results using data collected from a driver-assistance test-bed.

Index Terms—Autonomous systems, human-in-the-loop control, machine learning, modeling.

I. INTRODUCTION

HUMAN–ROBOT collaboration (HRC) studies how to achieve effective collaborations between human and robots to synthetically combine the strengths of human beings and robots. While robots have advantages in handling repetitive tasks with high precision and long endurance, human beings are much more flexible to changing factors or uncertain environments that are difficult for robots to adapt. Therefore, to establish an efficient collaboration between human and robots is the core problem in the design of the HRC system.

To achieve an effective HRC, it is of critical importance to obtain an expressive but meanwhile tractable model for HRC. Among several types of models in the HRC literature, such as the Adaptive Character of Thought-Rational/Embodied model [1], the Intention-Driven Dynamics Model [2] and the teaching-learning-prediction model [3], the partially observable Markov decision process (POMDP) has emerged as a popular choice in recent years [4], [5]. As a general probabilistic system model to capture uncertainties from actuation errors, sensing noises and human behaviors, the POMDP model provides a comprehensive framework for the system modeling and sequential decision making. Most of the existing results assume that the POMDP model is given [6], [7] or the number of states is given [8], [9] before learning the model from data. However, the state space could be tedious to be predefined and the number of hidden states could be case dependent especially when human is involved. In our previous work [10], we dropped these assumptions and proposed a Bayesian non-parametric learning approach to infer the structure of the POMDP model, such as the number of states, from data. However, hidden states of the POMDP model being learned can only model static properties of the observed data. For example, in the driving scenario, each state is a cluster of positions of the human hand. Then the human intention, say turning right, can be inferred if the observed position belongs to the cluster. Dynamic properties such as turning speed and acceleration could not be modeled and distinguished. This is because we did not consider the correlations among observations in the POMDP model.

In order to fill this gap, we propose a new type of model for HRC, called vector autoregressive partially observable Markov decision process (VAR-POMDP), which takes the observation correlation into consideration and hence extends the existing POMDP model. Our main objective in this letter is to show that the proposed VAR-POMDP model can achieve a good trade-off between model expressiveness and tractability.

The expressiveness of the proposed model is clear as the POMDP model becomes a special case of our proposed model. To illustrate the tractability of the proposed model, we investigate both the model learning and planning issues in this letter. First, in the model learning process, we do not assume the state space is given or the bound on the number of states is known. Our basic idea is to use a Bayesian non-parametric learning method to automatically identify the number of hidden states. Secondly, in the planning process, we consider the probabilistic computation tree logic (PCTL) as a formal specification since it is widely used as a safety
and reachability requirement in robotic applications [11]. The PCTL bounded until model checking problem can be converted to a finite horizon dynamic programming problem. We show that the value function can be approximated by a piece-wise linear function by extending the point-based value iteration (PBVI) on the VAR-POMDP model. The effectiveness of the proposed learning and planning algorithms are illustrated in real experiments.

The main contribution of this letter is twofold. First, the VAR-POMDP model which considers the correlations of observations is inspired from the autoregressive hidden Markov model (AR-HMM) which is an extension of the HMM. Distinct from the HMM which assumes the independence of observations, the autoregressive model specifies that the observed variable depends linearly on its previous values with certain uncertainties. This correlation property exhibits in the behavior of human motion [12]. Thus, we consider the correlation of observations in the POMDP model and propose the VAR-POMDP model.

Definition 1: The VAR-POMDP model is defined as a tuple 
\[ P = (S, A, O, T, E, R, L) \]
where
- \( S \) is a finite set of states,
- \( A \) is a finite set of decision actions,
- \( O \) is a set of continuous observations,
- \( T : S \times A \times S \to [0, 1] \) is a transition function which defines the probability over the next state \( s' \) after taking an action \( a \) from the state \( s \),
- \( E : O \times S \to \mathbb{R} \) is an observation function which defines the distribution over the observation \( o' \) that may occur in state \( s' \) conditional on observation history \( o \),
- \( R : S \times A \to \mathbb{R} \) is a reward function,
- \( L : S \to 2^{AP} \) is a labelling function that assigns a subset of atomic propositions \( AP \) to each state \( s \).

The difference between the VAR-POMDP model and the traditional POMDP model is the observation function. In the traditional POMDP model, the observation function only depends on the hidden state and the current observation. While in VAR-POMDP model, the observation function also relies on the observation history, namely,
\[ o_t = \sum_{i=1}^{r} A_{i,s_t} o_{t-i} + e(s_t), \]
where \( e(s_t) \sim N(0, \Sigma_{s_t}) \) are Gaussian noise modeling the uncertainty, matrices \( \{A_{1,s_t}, \ldots, A_{r,s_t}\} \) are lag matrices under mode \( s_t \). Note that the continuous observation POMDP with Gaussian emissions is a special case of this model when lag matrices are zero and the Gaussian mean is replaced with a constant vector. Instead of using a constant value to characterize the motion feature, the VAR-POMDP uses lag matrices and the covariance matrix to characterize the feature. Thus, dynamic features can be expressed and identified by the VAR-POMDP.

A. Probabilistic Computation Tree Logic

To achieve a performance-guaranteed model learning and planning framework, we use the PCTL as the formal specification to guide the designing process. PCTL is a probabilistic extension of computation tree logic which allows for probabilistic quantification of described properties [13]. The syntax of PCTL is as follows:
\[
\phi ::= \text{true} \mid \neg \phi \mid \phi_1 \land \phi_2 \mid P_{\leq p} \psi,
\]
\[
\psi ::= X\phi \mid \phi_1 \cup^{\leq k} \phi_2 \mid \phi_1 \cup \phi_2,
\]
where \( AP \) is an atomic proposition, \( \leq \in \{\leq, <, >, \} \), \( p \in [0, 1] \), \( k \in \mathbb{N} \) and \( \cup^{\leq k} \) stands for bounded until. The soft deadline property makes PCTL a widely used specification language for probabilistic model checkers.

In this letter, we consider the problem of VAR-POMDP model learning and planning for the bounded until specification in PCTL.

Problem 1: Given training data collected from an HRC process, learn a VAR-POMDP model \( P \). Based on the learned model \( P \), together with a given initial belief \( b_0 \) over the state, a finite horizon \( H \) and a PCTL bounded until specification,
\[
P_{\leq p} [\phi_1 \cup \leq H \phi_2],
\]
check whether or not the specification is satisfied.

The PCTL specification specifies the upper bound of the probability that, in a finite \( H \) step, there are some states along a path making \( \phi_2 \) holds and \( \phi_1 \) holds in all states prior to that state. For example, if \( \phi_1 = \text{true} \) and \( \phi_2 = \text{‘Fail’} \), the specification bounds the probability of the system going to states that cause the system failure. The specification in equation (3) is satisfied if and only if \( p\phi_1\phi_2^\text{max}(\phi_1 \cup \leq H \phi_2) \leq p \) where \( p\phi_1\phi_2^\text{max}(\phi_1 \cup \leq H \phi_2) \) is the maximum satisfaction probability with respect to belief \( b_0 \). The model checking problem is converted to a finite step optimization problem.

On one hand, the PCTL specification gives a performance requirement of the system which guides the model learning and controller designing process. On the other hand, using PCTL as specification avoids further defining the reward function for the VAR-POMDP model. Although algorithms such as inverse reinforcement learning can be used to recover the reward function [14], it is hard to explain the physical meaning of the reward. For PCTL specification, the reward can be clearly explained as satisfaction probability.

III. VAR-POMDP Model Learning

The proposed framework to learn the VAR-POMDP model is shown in Figure 1. The action set is assumed to be given since it represents the capability of the robot. Using the training data collected from the HRC system, the state space of the model is identified using the Bayesian non-parametric learning method. The whole state space could be a product of the state

...
space of human, robots and the environment. Based on the identified state space, transition probability and observation distribution can be learned from data.

A. Motion Feature Extraction

Instead of assuming that the state space is given, we use Bayesian non-parametric learning methods to directly infer the state space from data. The training data consists of several $n$-dimensional time series which could be human motion trajectories collected from demonstrations. Taking advantages of the beta process autoregressive HMM (BP-AR-HMM) framework proposed in [15], we do not only learn features of the trajectories collected from demonstrations. Taking advantages of Bayesian non-parametric learning methods to directly infer the distribution can be learned from data.

A driver and hardware-in-the-loop simulation system is used as an example to validate the proposed approach. Markers are put on the left/right hand of the driver and the steering wheel, a time series of positions of these markers are collected using the Optitrack system. An example of the raw data is shown in Figure 2 which consists of driving motions such as turning left and right. Using the Bayesian non-parametric learning method, motion features can be identified automatically. Figure 3 gives a comparison of learning results using HMM and AR-HMM as generative models for the same training data. Different motion features are labeled by different colors. From the result, there are only 3 motion features detected using the HMM generative model while there are 53 motion features detected using the AR-HMM generative model, which is much more than that of the HMM assumption. To give a more detailed comparison, data points from 1 to 1000 are zoomed out which is shown in Figure 4. From the results, some dynamic motions are not identified by the HMM model while they are detected using the AR-HMM. The reason behind this phenomenon is that the AR-HMM uses a dynamic system to model the observed data and considers the correlation among observations while HMM assumes observation independence.

B. Construct VAR-POMDP Model

Based on the features identified in Section III-A, the VAR-POMDP model can be directly constructed. First, the state space $S$ of the VAR-POMDP model is defined as the product of the state space of human, robots and the environment. The state space of the human can be defined as the union of motion features $\theta_j$ identified in Section III-A. Each state can be labeled manually with a physical meaning. According to the physical meaning, the labeling function can be defined. For example, in a driving scenario, the label could be ‘safe’, ‘danger’ or ‘failure’. The observation space $O$ is defined to be

$$B | B_0 \sim \text{BP}(c, B_0),$$

$$X_i | B \sim \text{BeP}(B),$$

$$\pi_j^i | f_i, \gamma, \kappa \sim \text{Dir}(\gamma, \ldots, \gamma + \kappa, \gamma, \ldots), \otimes f_i,$$
the $n$-dimensional vector space $\mathbb{R}^n$ which could be continuous sensor readings. The observation function $E$ is defined as a multivariate Gaussian distribution,

$$E(o_t|o_{t-1}, s_t) \sim N(\sum_{j=1}^r A_{j,t}s_t o_{t-j}, \Sigma_{s_t}),$$

where $o_{t-1}$ is the observation history.

After identifying the state space and observation distribution, our next step is to learn the transition probability. To learn the exact transition probability is difficult due to reasons such as limited data. In this case, modeling uncertainties will make the learned transition probabilities subject to a certain confidence level which motivates us to apply the Chernoff bound to reason the accuracy of the transition probabilities for VAR-POMDP. Details of the transitions probability learning can be found in [10]. From the Chernoff bound, the estimation error of the transition probability can be sufficiently small with high confidence as long as the training data is sufficient enough.

**Example 2:** In this example, parts of the model are shown in Figure 5 to illustrate the model construction process. For the driver assistance system, actions are designed to increase car safety and road safety by providing real-time instructions, warnings or directly controlling the vehicles. For example, one action $a_1$ can be designed as an instruction ‘No right turn’ and another action $a_2$ can be designed to increase the steering torque. Different actions have different influence on the human driver and therefor have different transition probability. Each state is a composition of environment status and human intentions. State $s_0$ represents $(on\_lane, turn\_right)$, $s_1$ represents $(on\_lane, Normal\_driving)$ and $s_2$ represents $(off\_lane, \times)$ where $\times$ can be any human intentions. Based on observations, a belief is maintained over hidden states according to the Bayes’ rule and an action can be selected to maximize the safety of the system.

**IV. VAR-POMDP PLANNING**

In the previous section, we showed that the proposed VAR-POMDP model can be effectively learned from data. This section aims to illustrate that the planning problem based on the proposed VAR-POMDP model is also tractable. Particularly, we consider PCTL specification as safety and reachability requirement and apply PBVI to solve the PCTL model checking problem.

**A. Converting the PCTL Model Checking Into a Dynamic Programming Problem**

We first show that the bounded until specification model checking problem described in Problem 1 can be converted into a dynamic programming problem. The PCTL model checking problem was studied for MDP model [17], we generalize the result to the VAR-POMDP model. The state space of the model can be divided into three disjoint subsets.

$$S^{yes} = \text{Sat}(\phi_2),$$
$$S^{no} = S \setminus (\text{Sat}(\phi_1) \cup \text{Sat}(\phi_2)),$$
$$S^? = S \setminus (S^{yes} \cup S^{no}).$$

All states in $S^{yes}$ satisfy $\phi_2$ and all states in $S^{no}$ dissatisfy both $\phi_1$ and $\phi_2$. The state in $S^?$ satisfies $\phi_1$ but not $\phi_2$. Once the system runs to states in $S^{no}$, the satisfaction probability will be zero no matter where it goes in the future. And once the system runs to states in $S^{yes}$, the satisfaction probability depends only on the running prefix. Thus changing the states in $S^{no}$ and $S^{yes}$ absorbed and assigning terminal reward 1 for states in $S^{yes}$ and 0 for other states does not change the satisfaction probability. The maximum satisfaction probability can
be solved recursively by value iteration,

\[ V_{t+1}(b, o) = \max_{a \in A} \{ \sum_{s, s'} b_s \int_{\phi} T^{a,s}_{s'} E^{o,a}_{s'} V_t(b', o') do' \}. \]  

(6)

where \( b_s \) stands for the belief on state \( s \), \( T^{a,s}_{s'} \) and \( E^{o,a}_{s'} \) are notations for transition probability and observation distribution, \( b' \) is the posterior belief after observing \( o' \) and \( o \) represents observation history. The value \( V_t(b, o) = p^{\text{max}}_b(\phi_1 \cup \mathbb{Z}^{\geq 2} \phi_2) \) is the maximum probability that satisfies the specification when the belief is \( b \) and observation history is \( o \). The initial condition \( p^{\text{max}}_b(\phi_1 \cup \mathbb{Z}^{\leq 0} \phi_2) = \sum_s b_s \rho^0_b \) where \( \rho^0_b = 1 \) if \( s \in \mathcal{S}^{\text{res}} \) and \( \rho^0_b = 0 \) otherwise.

### B. Point-Based Value Iteration for VAR-POMDP

The main challenge arising from the VAR-POMDP model is the curse of dimensionality and the curse of history. In equation (6), it is impossible to enumerate observations \( o' \) since \( o' \) is in continuous space. The value function is not only a function of belief \( b \) but also a function of observation history \( o \). Thus the exact dynamic programming approach cannot be applied to solve the problem [18]. Inspired by the dynamic discretization approach [19], we propose to use the PBVI to solve the dynamic programming problem.

In PBVI, a set of belief points \( \{\tilde{b}^1, \ldots, \tilde{b}^M\} \) is selected and the value function is updated only on these belief points. Thus PBVI gives an approximate solution where the approximation error dependents on the belief point selection.

**Theorem 1:** The optimization problem defined recursively by equation (6) can be solved using PBVI algorithm on a predefined belief set \( \{\tilde{b}^1, \ldots, \tilde{b}^M\} \). At these belief points, the value function \( V_t(b, o) \) can be approximated by a piece-wise linear and convex function, which is written as

\[ V_t(b, o) = \max_k \sum_s b_s \alpha^{t,k}_s, \]  

(7)

for a set of \( \alpha \)-vector \( \{\alpha^{1,1}, \ldots, \alpha^{t,M}\} \) where \( \alpha^{t,k}_s \) represents the \( k \)-th element of the \( s \)-th \( \alpha \)-vector for the \( t \)-step-to-go value function.

**Proof:** The theorem is proved by induction. Assume that the value function can be expressed as \( V_t(b, o) = \max_k \sum_s b_s \alpha^{t,k}_s \). When \( t = 0 \), the initial \( \alpha \)-vector \( \alpha^{0,1} \) is defined corresponding to the reward of the reaching state. According to the \( 0 \)-step-to-go value function defined in Section IV-A, the initial \( \alpha \)-vector \( \alpha^{0,1} = \rho^0_b \) and it is a constant vector. When \( t = 1 \), substitute the updated belief

\[ \hat{b}_s' = \frac{E^{o,0}_s \sum_{s'} T^{a,s}_{s'} b_{s'}}{\sum_s E^{o,0}_s \sum_{s'} T^{a,s}_{s'} b_{s'}}, \]  

(8)

and value function (7) into the right hand of equation (6),

\[ V_1(b, o) = \max_{a \in A} \{ \sum_s b_s \int_{\phi} T^{a,s}_{s'} E^{o,a}_{s'} \alpha^{0,1}_s do' \} \]

\[ = \max_{a \in A} \{ \sum_s b_s \sum_{s'} T^{a,s}_{s'} \alpha^{0,1}_s \}. \]  

(9)

For each point in the belief set \( \{\tilde{b}^1, \ldots, \tilde{b}^M\} \), we select the action \( a \) that maximizes the expected reward. Then a set of \( \alpha \)-vector \( \{\alpha^{t,k}\}, k \in \{1, \ldots, M\} \) is updated.

Assume for all \( t \geq 2 \), the value function can be expressed as \( V_{t-1}(b, o) = \max_k \sum_s b_s \alpha^{t-1,k}_s \). Then the \( t \)-step-to-go value function can be expressed as

\[ V_t(b, o) = \max_{a \in A} \{ \int_{\phi} \max_k \sum_s b_s T^{a,s}_{s'} E^{o,a}_{s'} \alpha^{t-1,k}_s do' \}. \]  

(10)

Due to the max function inside the integral, directly calculating the integration is not tractable. Inspired by the work of [19], we break the observation space into sub-space and use the sampling-based approach to approximate the integration.

Let \( z_k(o) \) be the sub-space of the observation \( o' \) that makes the expected reward maximum for a specific belief point and \( \alpha \)-vector. For a given belief \( \tilde{b} \) and \( \alpha \)-vector \( \alpha^{t-1,k} \),

\[ z_k(o) = \{o' | \max \{ \sum_s \hat{b}_s' \sum_{s'} T^{a,s}_{s'} E^{o,a}_{s'} \alpha^{t-1,k}_s \} = k \}. \]  

(11)

Then the value iteration is converted to

\[ V_t(b, o) = \max_{a \in A} \{ \sum_k \sum_s \hat{b}_s' \sum_{s'} T^{a,s}_{s'} E^{o,a}_{s'} \alpha^{t-1,k}_s \int_{z_k(o)} E^{o,a}_{s'} do' \} \]

\[ = \max_{a \in A} \{ \sum_k \sum_s \hat{b}_s' \sum_{s'} T^{a,s}_{s'} E^{o,a}_{s'} \int_{z_k(o)} \frac{E^{o,a}_{s'} do'}{E^{o,a}_{s'} do'} \} \]

\[ = \max_{a \in A} \{ \sum_k \sum_s \hat{b}_s' \sum_{s'} T^{a,s}_{s'} E^{o,a}_{s'} \int_{z_k(o)} \frac{E^{o,a}_{s'} do'}{E^{o,a}_{s'} do'} \} \]

\[ = \max_{a \in A} \{ \sum_k \sum_s \hat{b}_s' \sum_{s'} T^{a,s}_{s'} E^{o,a}_{s'} \int_{z_k(o)} \frac{E^{o,a}_{s'} do'}{E^{o,a}_{s'} do'} \} \]

\[ \int_{z_k(o)} \frac{E^{o,a}_{s'} do'}{E^{o,a}_{s'} do'} \]  

(12)

where \( \Pr(z_k(o)|s') = \int_{z_k(o)} E^{o,a}_{s'} do' \). To calculate the integral directly is not tractable, thus sampling approach is used to approximate the integration.

The sub-space \( z_k(o) \) currently is a function of observation history \( o \) which causes the sampling from \( E^{o,a}_{s'} do' \) intractable. However, it can be shown that the probability \( \Pr(z_k(o)|s') \) does not depend on observation history \( o \).

Let \( \hat{o} = o' - \sum_{r=1}^{t} A_j s \alpha_{t-r} \) be a new variable for which the observation distribution is \( z_k(o) \). Then

\[ \Pr(z_k(o)|s') = \int_{z_k(o)} E^{o,a}_{s'} do' = \int_{z_k(o)} E^{o,a}_{s'} do'. \]  

(13)

The max function of equation (15) is evaluated on belief set \( \{\tilde{b}^1, \ldots, \tilde{b}^M\} \). By induction, we conclude that the value function \( V_t(b, o) \) can be approximated by a piece-wise linear and convex function for all \( t \geq 1 \).

From the proof, it is shown that \( M|A| \) \( \alpha \)-vectors are created at each time step while the number of final \alpha-vectors stored is limited to \( M \) (in time complexity \( M^2|S|^2|A| \)). Thus, the whole value iteration takes polynomial time and the size of \( \alpha \)-vectors remains constant. The PBVI is an approximation
Algorithm 1 PBVI Algorithm for VAR-POMDP

Input: Vector set \( \{\alpha^i_1, ..., \alpha^i_M\}, i \in \{1, ..., H\}\).

Output: Belief set \( \{\hat{b}^1, ..., \hat{b}^M\}\).

1. Initialize \( \alpha\)-vector \( \alpha^0 = p^0 \).

2. while \( t \leq H \) do

3. \hspace{0.5cm} if \( t = 1 \) then

4. \hspace{1cm} for \( \hat{b}^i \in \{\hat{b}^1, ..., \hat{b}^M\} \) do

5. \hspace{1.5cm} Update \( \alpha\)-vector according to equation (9).

6. \hspace{0.5cm} end for

7. \hspace{0.5cm} else

8. \hspace{1cm} for \( \hat{b}^i \in \{\hat{b}^1, ..., \hat{b}^M\} \) do

9. \hspace{1.5cm} Approximate \( Pr(z_t|s') \).

10. \hspace{1.5cm} Update \( \alpha\)-vector according to equation (15).

11. \hspace{0.5cm} end for

12. \hspace{0.5cm} end if

13. \hspace{0.5cm} \( t = t + 1 \).

14. end while

\begin{center}
\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure6.png}
\caption{The transition relation of a three-state VAR-POMDP model.}
\end{figure}
\end{center}

algorithm, it scarifies the accuracy of the solution to achieve efficiency. The approximation error depends on how densely the belief set \( \{\hat{b}^1, ..., \hat{b}^M\} \) samples from the belief simplex \( \Delta \). Following the proof of [20, Lemma 1] with the fact that the maximum value is 1 in our scenario, it can be shown that the approximation error is bounded by the density of the belief set. Due to the space limitation, an extended version is provided with a formal proof [21]. The value iteration algorithm is summarized in Algorithm 1. Since it is a finite-step value iteration, the algorithm will always converge.

**Example 3:** A three-state VAR-POMDP model shown in Figure 6 is used to validate the PBVI algorithm. The transition probability of action \( a_1 \) and \( a_2 \) are shown in the figure. Each state represents a motion feature of a three-dimensional motion trajectory and state \( s_2 \) are labeled as ‘Fail’. The belief points are selected to be \( \{1, 0, 0\}, \{0, 1, 0\}, \{0.5, 0.5, 0\}, \{0.6, 0.3, 0.1\}, \{0.3, 0.4, 0.3\} \). The PCTL specification is given as \( P_{\leq 0.5}[\phi_1 U_{\leq 2} \phi_2] \) with \( \phi_1 = true \) and \( \phi_2 = \lnot \phi \). Using the PBVI, five \( \alpha\)-vectors are solved which are \( [0.67, 0.63, 1], [0.63, 0.71, 1], [0.64, 0.71, 1], [0.64, 0.71, 1] \). If the initial belief is \( \{1, 0, 0\} \), then the satisfaction probability is 0.67. Since the upper bound is 0.5, the specification is not satisfied.

**V. CONCLUSION**

In this letter, we proposed the VAR-POMDP model for HRC which is an extension of the traditional POMDP model by considering the correlation among observations. We showed the tractability of the proposed model by providing a learning framework and a planning algorithm. In the learning framework, we proposed to use Bayesian non-parametric methods to learn the VAR-POMDP model from demonstrations effectively. We proved that the PBVI algorithm can be extended on the VAR-POMDP model to solve a model checking problem for bounded until specification in PCTL. In both the learning and planning process, approximations were used to estimate parameters of the model including the transition probability, observation distribution and the potential belief points. Evaluating the influence of these approximations on system performance will be our future work.

**REFERENCES**

[1] G. Trafton, L. M. Hiatt, A. M. Harrison, F. P. Tamborello, II, S. S. Khemlani, and A. C. Schultz, “ACT-R/E: An embodied cognitive architecture for human–robot interaction,” J. Human Robot Interact., vol. 2, no. 1, pp. 30–55, 2013.

[2] Z. Wang et al., “Probabilistic movement modeling for intention inference in human–robot interaction,” Int. J. Robot. Res., vol. 32, no. 7, pp. 841–858, 2013.

[3] W. Wang, R. Li, Y. Chen, and Y. Jia, “Human intention prediction in human–robot collaborative tasks,” in Proc. Companion ACM/IEEE Int. Conf. Human Robot Interact., 2018, pp. 279–280.

[4] E. B. Fox, M. I. Jordan, E. B. Sudderth, and A. S. Willsky, “Sharing features among time series via the beta process with applications to the joint inference of multiple time series,” J. Mach. Learn. Res., vol. 15, no. 55, pp. 1225–1232, Jun. 2019.

[5] X. Zhang and H. Lin, “Performance guaranteed human–robot collaboration with POMDP supervisory control,” Robot. Comput. Integ. Manuf., vol. 57, pp. 59–72, Jun. 2019.

[6] F. Broz, I. Nourbakhsh, and R. Simmons, “Designing POMDP models of socially situated tasks,” in Proc. IEEE RO MAN, 2011, pp. 39–46.

[7] N. Gopalan and S. Tellex, “Modeling and solving human–robot collaborative tasks using POMDPs,” in Proc. Robot. Sci. Symp., 2015, pp. 1–7.

[8] J. J. M. Rutten, “Acting in partially observable Markov decision processes,” in Proc. Eur. Conf. Mach. Learn., 2005, pp. 601–608.

[9] S. Ross, B. Chai-Draa, and J. Pineau, “Bayes-adaptive POMDPs,” in Proc. Adv. Neural Inf. Process. Syst., 2008, pp. 1225–1232.

[10] W. Zheng, B. Wu, and H. Lin, “POMDP model learning for human robot collaboration,” in Proc. IEEE Conf. Decis. Control (CDC), 2008, pp. 1156–1161.

[11] X. Zhang, B. Wu, and H. Lin, “Learning based supervisor synthesis of POMDP for PCTL specifications,” in Proc. 54th IEEE Conf. Decis. Control (CDC), 2015, pp. 7470–7475.

[12] K. Abe, H. Miyatake, and K. Oguri, “A study on switching AR-HMM driving behavior model depending on driver’s states,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), 2007, pp. 806–811.

[13] E. B. Fox, M. C. Hughes, E. B. Sudderth, and M. I. Jordan, “Joint inference of multiple time series via the beta process with applications to the joint inference of multiple time series,” J. Mach. Learn. Res., vol. 8, no. 3, pp. 1281–1313, 2014.

[14] J. Choi and K.-E. Kim, “Inverse reinforcement learning in partially observable environments,” J. Mach. Learn. Res., vol. 12, pp. 691–730, Mar. 2011.

[15] W. Wang, R. Li, Y. Chen, and Y. Jia, “Human intention prediction in human–robot collaborative tasks,” in Proc. Companion ACM/IEEE Int. Conf. Human Robot Interact., 2018, pp. 279–280.

[16] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for POMDPs,” in Proc. IJCAI, pp. 841–858, 2013.

[17] H. Hansson and B. Jonsson, “A logic for reasoning about time and reliability,” J. Mach. Learn. Res., vol. 3, no. 7, pp. 860–881, 2002.

[18] J. J. M. Rutten, “Acting in partially observable Markov decision processes,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 549–557.

[19] J. Choi and K.-E. Kim, “Inverse reinforcement learning in partially observable environments,” J. Mach. Learn. Res., vol. 12, pp. 691–730, Mar. 2011.

[20] E. J. Jondadbekian, “The optimal control of partially observable Markov processes,” Stanford Electron. Lab., Stanford Univ., Stanford, CA, USA, Rep. SU-SEL-71-017, 1971.

[21] J. Fox, M. I. Jordan, E. B. Sudderth, and A. S. Willsky, “Sharing features among dynamical systems with beta processes,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 1332–1338.