An Adaptive Nonparametric Particle Filter for State Estimation

Yali Wang and Brahim Chaib-draa

Abstract— **Particle filter is one of the most widely applied stochastic sampling tools for state estimation problems in practice. However, the proposal distribution in the traditional particle filter is the transition probability based on state equation, which would heavily affect estimation performance in that the samples are blindly drawn without considering the current observation information. Additionally, the fixed particle number in the typical particle filter would lead to wasteful computation, especially when the posterior distribution greatly varies over time. In this paper, an advanced adaptive nonparametric particle filter is proposed by incorporating gaussian process based proposal distribution into KLD-Sampling particle filter framework so that the high-qualified particles with adaptively KLD based quantity are drawn from the learned proposal with observation information at each time step to improve the approximation accuracy and efficiency. Our state estimation experiments on univariate nonstationary growth model and two-link robot arm show that the adaptive nonparametric particle filter outperforms the existing approaches with smaller size of particles.**

I. INTRODUCTION

State estimation refers to a filtering problem which uses the observation history to infer the underlying hidden state. It has been widely used for real applications in science and engineering, such as economics, biostatistics, signal processing, robotics, etc. Because the foundation of understanding dynamic systems is to estimate its true state over time, an accurate and efficient filtering technique is highly crucial for dynamic phenomena analysis. For example, a mobile robot attempts to find where it is via state estimation if it wants to finish the high level tasks like path planning in the real environment [1].

During the past decades, there have been a great number of stochastic filtering algorithms. One of the most famous technique is Kalman filter (KF) [3]. KF is the best-known optimal filter for linear systems with Gaussian noises. But it is relatively theoretical since most real applications are nonlinear and/or nongaussian. Hence, many extensions including extended kalman filter [4] and unscented kalman filter [5] have been investigated to efficiently estimate the hidden state. However, all those extensions approximate the posterior distribution as a gaussian distribution using KF framework, so they terribly deteriorate the estimation performance when the true posterior is highly nongaussian.

A class of particle filters based on sequential monte carlo sampling method [6] [7] [8] is an alternatively effective way to approximate the arbitrary nongaussian posterior by recursively stochastic sampling. The typical sampling importance resampling (SIR) particle filter consists of two steps - important sampling and resampling at each time to estimate the important region of posterior [7]. However, SIR particle filter suffers weight degeneracy problem: the variance of the particle weights will be very large after a few iterations [7], which means that most particles are wasteful with very small weight. The resampling step hides this problem by replicating the particles with large weight, introducing thus a high correlation between particles. In fact, the root cause of weight degeneracy problem is that particles drawn from the designed proposal distribution are located in the low probabilistic regions of true posterior [10]. Hence, choosing a better proposal distribution should be done firstly. In order to approximate the true posterior as close as possible, extended kalman particle filter and unscented particle filter have been developed by local linearization [11]. However, there exists a severe limitation of parametric methods that the density model specified by a few parameters lacks of robustness, especially when the dynamic system is high-dimensional and nongaussian. Bayesian nonparametric technique is a promising strategy that could tackle the estimation problem for more general dynamic systems. Some researchers as [12] [14] [15] proposed a gaussian process based particle filter for robotics. However, the fixed particle number results in the approximation with low computational efficiency, especially when true posterior greatly changes over time. A general particle filter framework for this problem is called KLD-Sampling particle filter [13], which adaptively adjusts the particle number by Kullback-Leibler divergence to improve the estimated performance.

In this paper, we present an adaptive nonparametric particle filter to efficiently solve state estimation problem. The proposed algorithm combines gaussian process based proposal distribution into KLD-Sampling particle filter framework. Firstly, the approximately optimal proposal distribution incorporates current observation information so that the particles would be more likely located at the important region of true posterior distribution. Then, the necessary particles would be drawn by KLD-Sampling at each time step to improve the performance. The outline of this paper is as follows: firstly, we review the state space model and traditional SIR particle filter in Section II. Then a gaussian process based particle filter and KLD-Sampling particle filter will be introduced in Section III and Section IV to deal with the drawbacks in SIR particle filter. In Section V, we propose an adaptive nonparametric particle filter framework. After that, some experiments will be operated in Section VI. Finally, we conclude the paper in Section VII.

Y. Wang and B. Chaib-draa are with Department of Computer Science and Software Engineering, Laval University, Quebec, Quebec G1V 0A6, Canada wang@damas.ift.ulaval.ca; brahim.chaib-draa@ift.ulaval.ca

II. SAMPLING IMPORTANCE RESAMPLING PARTICLE FILTER

A. Dynamic System Modeling

The state space model for a nonlinear dynamic system with additive noises is defined:

$$\mathbf{x}_t = \mathbf{f}(\mathbf{x}_{t-1}) + \mathbf{w}_{t-1} \tag{1}$$

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{v}_t \tag{2}$$

where at time t, $\mathbf{x}_t$ is the hidden state vector, $\mathbf{w}_t$ is the system noise, $\mathbf{y}_t$ is the observation vector, $\mathbf{v}_t$ is the observation noise.

With the known distribution of $\mathbf{w}_t$ and $\mathbf{v}_t$, state equation (1) and observation equation (2) represent transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$. Additionally, it is assumed that state transition and observation process satisfy first-order Markovian, then state space model is transformed into hidden markov model which consists of initial density $p(\mathbf{x}_0)$, transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$. The filtering problem aims to evaluate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ and/or $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$ [2].

B. SIR Particle Filter

For a general filtering problem $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$, if the samples $\mathbf{x}_{0:t}^{(i)}$ $(i = 1, \cdots, N)$ could be generated from $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$, then we have Monte Carlo approximation:

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}) \tag{3}$$

where $\delta_{\mathbf{x}_{0:t}^{(i)}}(\cdot)$ is the dirac mass at $\mathbf{x}_{0:t}^{(i)}$. However, it is almost impossible to directly apply Monte Carlo sampling methods due to the unknown $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$. Hence, Sequential Monte Carlo Sampling procedure is introduced to iteratively estimate the subtask at each step, which produces a sampling importance resampling (SIR) particle filter [6] [7] [8].

The procedure of SIR particle filter is recursively done as follows: at time t, N particles are drawn from proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ that is designed as transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and then set $\mathbf{x}_{0:t}^{(i)} = \{\mathbf{x}_{0:t-1}^{(i)}, \mathbf{x}_t^{(i)}\}$ $(i = 1, 2, ...N)$. Secondly, particle weights are calculated and normalized by likelihood density $p(\mathbf{y}_t|\mathbf{x}_t)$. Finally, current N particles are resampled with their weights to obtain N equal-weighted new particles for the next step. Using the particles at time t, Monte Carlo approximation could be represented as:

$$\hat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \sum_{i=1}^{N} \tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)}) \delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}) \tag{4}$$

where importance weight $\mathbf{w}_t(\mathbf{x}_{0:t}^{(i)})$ is normalized to $\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})$.

Even though SIR particle filter has been widely used in the real applications [1] [18], there are two main drawbacks [10] [13]. Firstly, the resampling step just transforms particle degeneracy problem into another form - sample diversity impoverishment because of copying the important particles with large weights, but it does not solve weight degeneracy. The underlying cause is that the particles are not located at the important region of posterior distribution because the proposal distribution does not contain the new information from the observation $\mathbf{y}_t$. In fact, the weight update is

$$\begin{aligned} \mathbf{w}_t(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{x}_{0:t}, \mathbf{y}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})q(\mathbf{x}_{0:t-1}|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})}\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \end{aligned} \tag{5}$$

SIR particle filter simply applies the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ at each time step, which gives

$$\mathbf{w}_t(\mathbf{x}_{0:t}) = p(\mathbf{y}_t|\mathbf{x}_t)\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1}) \tag{6}$$

Since we do resampling at each time step, the weights at $t-1$ are the same which means that $\mathbf{w}_{t-1}(\mathbf{x}_{0:t-1})$ could be removed and (6) becomes:

$$\mathbf{w}_t(\mathbf{x}_{0:t}) = p(\mathbf{y}_t|\mathbf{x}_t) \tag{7}$$

But this simplification would lead to a SIR particle filter which is very sensitive for the outliers and it sequentially predicts the next state blindly, especially when the likelihood is relatively narrow. In figure 1, the high probability region of the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the low probability region of target distribution $p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Hence the samples that we blindly draw from proposal distribution are not located at the important region of target distribution, which causes that the weight distribution according to likelihood distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ does not approximate the target distribution. The fundamental reason is that the variance of likelihood distribution is usually relatively small, as shown in figure 1. It means that the current observation accurately describes the current state but the proposal distribution does not consider it. Hence, proposal distribution in

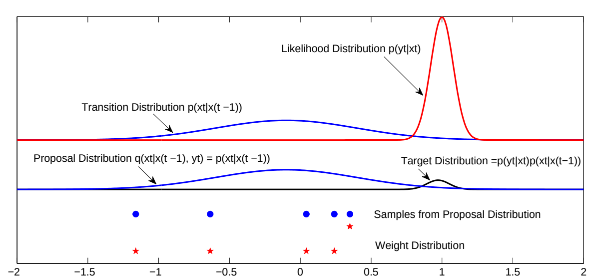

Fig. 1. Important Sampling at one step in Particle Filter. There is only 1 non-zero weight (rightmost star) in the weight distribution.

particle filter is one of the most crucial factors to determine whether its estimation is good or not.

The second problem is the fixed particle number in SIR particle filter. Since particle filter requires relatively high computation: time exponential in n where n is the dimension of the state vector, the effective number of particles is very important for computational efficiency, especially in high dimensional systems. The fixed number would cause high computation when the true posterior distribution varies vastly over time.

III. GAUSSIAN PROCESS BASED PARTICLE FILTER

In order to solve the problem of particle distribution in particle filter, one can apply the optimal proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)$ in the weight update. However, it is impossible to do so because this optimal proposal is actually unknown. We can therefore approximate it using density estimation methods. The parametric methods in general could not completely catch the dynamic phenomena in the complicated high-dimensional systems. In contrast, nonparametric methods are more robust to learn the distribution. Recently, a gaussian process based particle filter is proposed in [12] to learn the optimal proposal distribution to efficiently detect failure on mobile robots.

A. Gaussian Process Regression

A gaussian process is a set of random variables, any finite number of which has a joint gaussian distribution [16]. It is a nonparametric method which represents a gaussian distribution over functions.

A regression problem could be solved by the gaussian process as follows [14] [19]: Suppose the training set with m data instances is $Dataset = < X^{training}, \mathbf{y}^{training} >$, where the input matrix $X^{training} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]$ consists of n-feature input instances $\mathbf{x}_i$ (i = 1,2,...m), and $\mathbf{y}^{training} = [y_1, y_2, \cdots, y_m]$ is the output vector which is generated by

$$y_i = g(\mathbf{x}_i) + \varepsilon \tag{8}$$

g is a nonlinear function and $\varepsilon \sim N(0, \sigma_\varepsilon^2)$. Since the joint distribution of output variable vector $\mathbf{y}$ is a multi-variable gaussian distribution [17], given a test input $\mathbf{x}_*$, a predictive density over the target output y_* is specified as a conditional gaussian distribution according to the training set:

$$p(y_*|\mathbf{x}_*, Dataset) = N(y_*; \mu(\mathbf{x}_*, Dataset), \Sigma(\mathbf{x}_*, Dataset)) \tag{9}$$

with the mean

$$\mu(\mathbf{x}_*, Dataset) = \mathbf{k}_*^T [K + \sigma_\varepsilon^2 I]^{-1} \mathbf{y}^{training} \tag{10}$$

and the variance

$$\Sigma(\mathbf{x}_*, Dataset) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T [K + \sigma_\varepsilon^2 I]^{-1} \mathbf{k}_* \tag{11}$$

where $\mathbf{k}_*$ is a covariance vector that illustrates the kernel connection between the test input and the training set. K is the kernel matrix of training set. The kernel function k specifies the element for $\mathbf{k}_*$ and K: $\mathbf{k}_*[i] = k(\mathbf{x}_*, \mathbf{x}_i)$, $K[i,j] = k(\mathbf{x}_i, \mathbf{x}_j)$. One of the most popular kernel functions is gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_g^2 exp[-0.5(\mathbf{x}_i - \mathbf{x}_j) W (\mathbf{x}_i - \mathbf{x}_j)^T] \tag{12}$$

where σ_g^2 is the signal variance, W reflects the smoothness of the process [14]. Those hyperparameters σ_ε^2, σ_g^2 and W could be learned via maximum likelihood [16] [17].

B. SIR Particle Filter with Gaussian Process Proposal

The optimal proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)$ actually reflects the nonlinear relationship between $[\mathbf{x}_{t-1}, \mathbf{y}_t]$ and $\mathbf{x}_t$:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{x}_{t-1}, \mathbf{y}_t) + \varepsilon \tag{13}$$

where $\mathbf{g}$ is a nonlinear unknown function, $\varepsilon \sim N(\mathbf{0}, \sigma_\varepsilon^2 \mathbf{I})$.

This underlying relationship provides the possibility of applying gaussian process regression to learn the optimal proposal distribution. The density estimation is transformed as a model learning problem using gaussian process. The training set ($t = 1$ to T) is $Data = < [\mathbf{x}_{t-1}^{training}, \mathbf{y}_t^{training}], \mathbf{x}_t^{training} >$, then the gaussian process regression model is:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t) \sim \mathbf{N}(\mu([\mathbf{x}_{(t-1)}, \mathbf{y}_t], Data), \Sigma([\mathbf{x}_{(t-1)}, \mathbf{y}_t], Data)) \tag{14}$$

Applying gaussian process based proposal distribution into SIR particle filter results in the following algorithm:

- For $t = 0, 1, 2, ...T$
  1) For $i = 1, 2, ...N$:
     a) sample $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)$
     b) set $\mathbf{x}_{0:t}^{(i)} = \{\mathbf{x}_{0:t-1}^{(i)}, \mathbf{x}_t^{(i)}\}$
  2) For $i = 1, 2, ...N$: calculate the weights by
     $$\mathbf{w}_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)} \tag{15}$$
     and then normalize to $\tilde{\mathbf{w}}_t(\mathbf{x}_{0:t}^{(i)})$
  3) resample the current N particles by their weights to obtain N new particles with equal weights $1/N$

Notice the optimal proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{t-1},\mathbf{u}_{t-1})$ for real robot systems should contain control input $\mathbf{u}_{t-1}$ [12].

IV. KLD-SAMPLING PARTICLE FILTER

As previously stated, the necessary particle number, which is determined by the complexity of posterior density, highly affects the performance of state estimation. For example, if the posterior is highly nongaussian, we apparently need more particles to capture it, vice versa. However, SIR particle filter fixes the particle number, which results in a poor performance if the posterior distribution varies greatly over time. For improving computational efficiency, KLD-Sampling particle filter has been proposed by [13]. By using Kullback-Leibler divergence(KLD) to restrict estimated error, this approach adaptively regulates the particle number over time according to the posterior complexity:

$$KL[\hat{p}(\mathbf{x}) \| p(\mathbf{x})] = \sum_{\mathbf{x}} \hat{p}(\mathbf{x}) \log(\hat{p}(\mathbf{x})/p(\mathbf{x})) \tag{16}$$

where $\hat{p}(\mathbf{x})$ is the discrete estimation of $p(\mathbf{x})$. For any discrete $p(\mathbf{x})$ with k different bins, the number of samples from $\hat{p}(\mathbf{x})$ is computed as:

$$N_f = \chi_{k-1,1-\delta}^2 / 2\varepsilon \tag{17}$$

It ensures that KLD between the true density and its maximum likelihood estimation is smaller than ε with $1-\delta$ confidence. In the formula (17), χ_{k-1}^2 is the chi-square distribution with $k-1$ degrees of freedom.

KLD particle filter for the dynamic system in Section II:

1) Initialize ε and δ
2) For $t = 1, 2, \ldots T$, set $N = 0$, $k = 0$
   a) Sample a particle $\mathbf{x}_{t-1}^{(i)}$ with the normalized weights at $t-1$
   b) Sample a particle $\mathbf{x}_t^{(N)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$
   c) Calculate its weight according to $p(\mathbf{y}_t|\mathbf{x}_t^{(N)})$
   d) if $(\mathbf{x}_t^{(N)}$ falls into an empty bin $b)$ then
      i) $k = k+1$
      ii) Set b non-empty
   e) $N = N+1$
   f) if $N > N_f$, return $\mathbf{x}_t^{(i)}$ $(i = 1, 2, \ldots N)$. Otherwise, return to a)

For practical robot systems, particles are actually drawn from $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$ where $\mathbf{u}_{t-1}$ is control input [13]. Notice that the true posterior distribution is actually unknown for filtering problem, which leads to the unknown bin number k. Hence at each time step in this algorithm, k is incremental if a drawn particle falls into a new bin. Finally, KLD-Sampling particle filter obtains the necessary particle number N_t to estimate the true posterior distribution.

V. Adaptive Gaussian Proecess based Particle Filter

Gaussian process based particle filter alleviates the weight degeneracy problem via learning the optimal proposal distribution, but the particle number is always fixed. On the contrary, KLD-Sampling particle filter applies necessary particle number to approximate the posterior distribution but the particles are always drawn from transition probability without considering current observation information.

We now propose a novel hierarchical framework to combine these two methods together to take advantage of their merits so that we can improve the state estimation accuracy and efficiency. Firstly, an approximate optimal proposal is learned by nonparametric gaussian process so that the current observation information could be taken into account when drawing particles from proposal distribution. This step ensures that the sampled particles are more likely located at the high probabilistic region of true posterior distribution to improve the accuracy. Then, this trained proposal distribution is applied into KLD-Sampling particle filter framework to adaptively adjust the particle number for state estimation. This step guarantees that the correctly located particles could be drawn with the approximation requirement according to the complexity of true posterior distribution. Then the drawn particles have high quality and quantity to estimate the underlying hidden states.

VI. Numerical Illustration

A. Univariate Nonstationary Growth Model

Considering the classic univariate nonstationary growth model in [8]:

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + w_t \tag{18}$$

$$y_t = 0.05x_t^2 + v_t \tag{19}$$

where the system and observation noise are assumed as gaussian distribution $w_t \sim N(0, 10)$, $v_t \sim N(0, 1)$ respectively. The initial state distribution is $x_0 \sim N(0, 10)$. Time interval is set to 0.01 and the terminal time is 0.5, then the objective is $p(x_{0:50}|y_{0:50})$ that is high-dimensional in temporal space. The setting of the proposed method is as follows: both ε and δ are set to 0.05; for simplicity, the hyperparameters in the gaussian process are set: $\sigma_\varepsilon^2 = 1$, $\sigma_g^2 = 1$ and $W = I$. Finally, 50 training data at each step compose the training data set.

The simulations are shown by respectively running the proposed adaptive nonparametric particle filter and all other particle filters above. The result in figure 2 illustrates that estimation performance is very well using our approach since the approximated state is very close to the true state with high confidence.

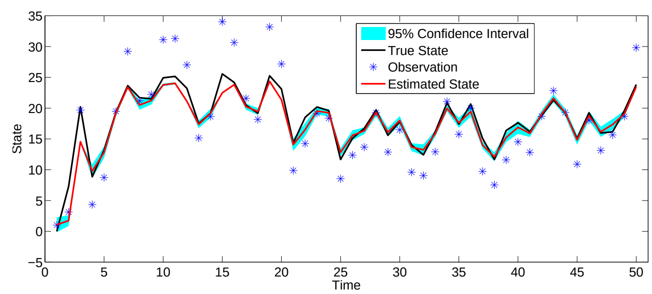

Fig. 2. State estimation by adaptive nonparametric particle filter

Using now root mean squared error (*RMSE*) as an evaluation criteria, we compare the filtering performance of different particle filters with different particle number in table I. Firstly, SIR particle filter with 10 particles ignores the current observation information since it uses transition probability as the proposal, and consequently, it has to increase the particle number to 100 in order to improve its accuracy. However, SIR particle filter is still inefficient even with 100 particles in that KLD-Sampling approach outperforms it only with average 45 particles. Then, our approach applies an observation-contained proposal in the KLD-Sampling framework instead of transition density, so it obtains a smaller *RMSE* than the traditional KLD-Sampling particle filter. Secondly, gaussian process based approach with 10 particles performs better than SIR particle with same particle number. The underlying reason is the learned proposal by gaussian process makes the particles located at more important region of true posterior because of considering current observation information. However, the 10-particle gaussian process based method is not good enough with comparison of our proposed method in that our approach applies KLD-Sampling to get the necessary particle number (average 45). Obviously, 10 particles in gaussian process based method is not adequate to completely capture true posterior. Above all, our proposed adaptive nonparametric method gives the best result with around 40 particles.

TABLE I
RMSE Comparison

Particle Filter (PF)	Particle Number	RMSE
SIR PF	10	2.0183
SIR PF	100	1.7209
KLD-Sampling PF	average 45	1.6373
Gaussian Process Based PF	10	1.5775
our Proposed PF	average 42	1.4348

B. Two-link Robot Arm

The two-link robot arm as proposed in [9] are shown in figure 3. Given the angles a_1 and a_2, the kinematic model of this robot is:

$$y_1 = r_1 cos(a_1) - r_2 cos(a_1 + a_2) \tag{20}$$

$$y_2 = r_1 sin(a_1) - r_2 sin(a_1 + a_2) \tag{21}$$

where $r_1 = 0.8$, $r_2 = 0.2$, the limited region are $a_1 \in [0.3, 1.2]$ and $a_2 \in [\pi/2, 3\pi/2]$, (y_1, y_2) is the cartesian position.

The goal here is to solve the inverse kinematic problem which estimates the angles a_1, a_2 given the observation (y_1, y_2) over time, and state space model of this two-link robot arm is

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{w}_{t-1} \tag{22}$$

$$\mathbf{y}_t = \begin{bmatrix} r_1 cos(a_{1,t}) - r_2 cos(a_{1,t} + a_{2,t}) \\ r_1 sin(a_{1,t}) - r_2 sin(a_{1,t} + a_{2,t}) \end{bmatrix} + \mathbf{v}_t \tag{23}$$

where state vector is $\mathbf{x}_t = [a_{1,t} \quad a_{2,t}]^T$, observation vector is $\mathbf{y}_t = [y_{1,t} \quad y_{2,t}]^T$, system noise is $\mathbf{w}_{t-1} \sim N(\mathbf{0}, diag\{0.02^2,\ 0.2^2\}$ and finally, the observation noise is $\mathbf{v}_t \sim N(\mathbf{0}, diag\{0.001^2,\ 0.001^2\})$. Thus, the measurement of the observations is very accurate.

The setting of this experiment is as follows: the initial state distribution is equal to the system noise. The robot arm moves 12 times, then the state estimation problem is specified as $p(\mathbf{x}_{0:12}|\mathbf{y}_{0:12})$. The setting of the proposed method is as follows: both ε and δ are set to 0.1; for simplicity, the hyperparameters are set $\sigma_\varepsilon^2 = 0.1$, $\sigma_g^2 = 0.1$ and $W = I$; 300 training data at each time step compose the training set.

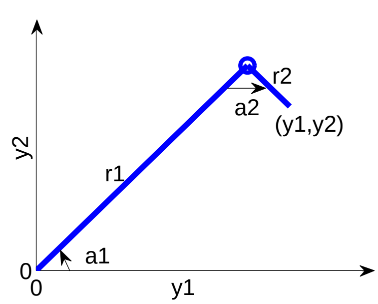

Fig. 3. Two-link Robot Arm

Using our proposed particle filter, we got the following experiment results. In figure 4 and 5, the true angles are well estimated.

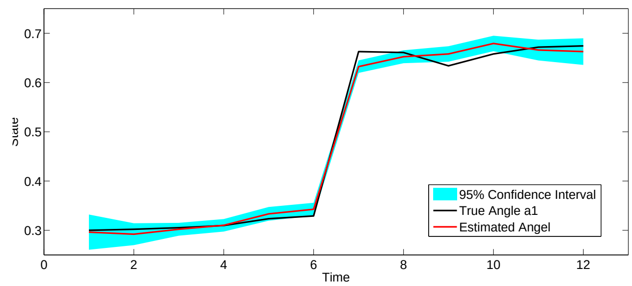

Fig. 4. a_1 estimation by adaptive nonparametric particle filter

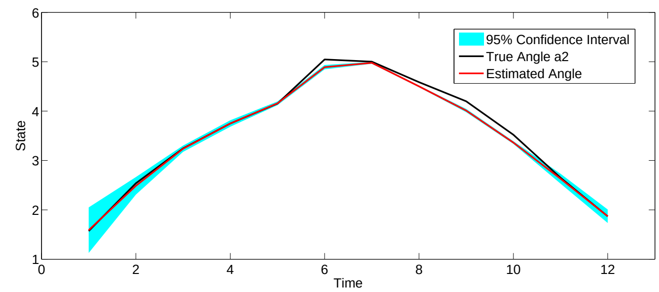

Fig. 5. a_2 estimation by adaptive nonparametric particle filter

Table II shows the performance comparison of different particle filters, where *RMSEi* is the *RMSE* of the angles a_i (i=1, 2). Firstly, SIR particle filter with 20 particles performs worst. There are two main reasons: (1) SIR approach does not consider the current observation information in its proposal. In this experiment, the observation measurement is highly accurate, which means that current observation could well represent the hidden state. However, SIR particle filter ignores it by using transition density as the proposal. This could lead that more drawn particles are actually located at the low probabilistic region of true posterior. This is why the performance of 20-particle SIR approach is worse than gaussian process based method with same particle number. (2) SIR approach fixes the particle number. From figure 6, fixed 20 particles in SIR method is not adequate since KLD-Sampling particle filter uses average 25 particles to get better estimation. The only way to get a smaller error for SIR approach is to increase the particle number to 150 with the computation load tradeoff. Secondly, gaussian process based method with smaller particle number gets better performance than KLD-Sampling approach. The fundamental reason is that the observation-contained proposal is more similar to the true posterior than the transition density. It causes that particles drawn from gaussian process based method could be more likely located at the important region of posterior than the ones from transition probability in KLD-Sampling approach. In order to overcome the particle location bias, KLD-Sampling approach adaptively increases the particle number. Finally, the proposed method combines both merits in KLD-Sampling and gaussian process based algorithms to get the best estimation with smaller size of particles.

Additionally, figure 7 shows that the true state and the

TABLE II
RMSE COMPARISON

Particle Filter (PF)	Particle Number	RMSE1	RMSE2
SIR PF	20	0.0378	0.1711
SIR PF	150	0.0207	0.1258
KLD-Sampling PF	average 25	0.0182	0.1140
Gaussian Process Based PF	20	0.0177	0.1085
Our Proposed PF	average 15	0.0154	0.0933

estimated state in this robot arm at each time to illustrate that the proposed algorithm is better than the tradition particle filter.

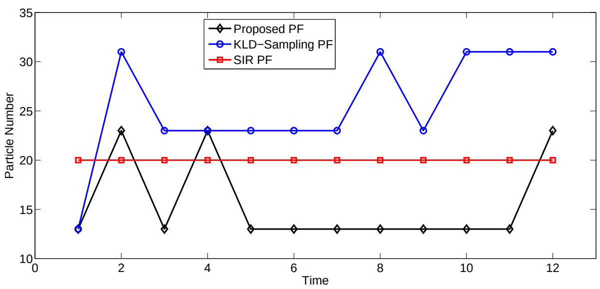

Fig. 6. Change of particle number over time

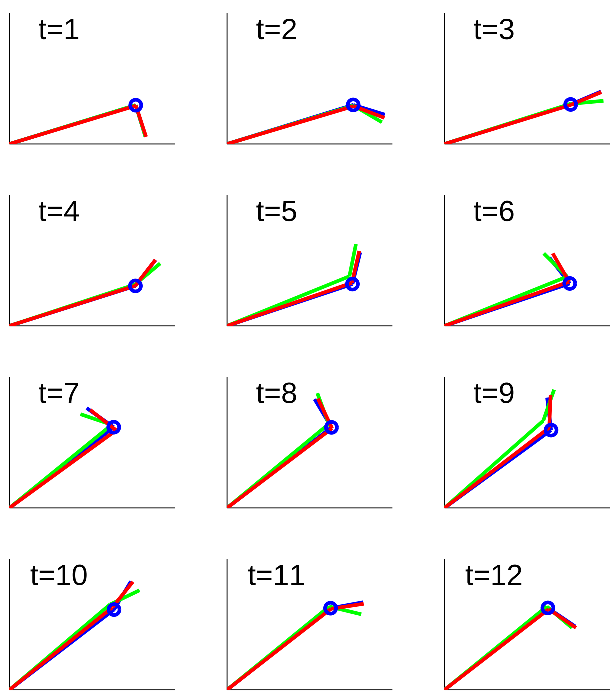

Fig. 7. Two-link robot arm position over time. The blue line with circle is the true state, the red line is estimated state by proposed method, the green line is estimated state by SIR particle filter

VII. CONCLUSION AND FUTURE WORK

This paper proposed an advanced adaptive nonparametric particle filter. Firstly, the particles drawn from the learned optimal proposal by gaussian process would more likely locate at the high probabilistic region of true posterior density in that the current observation information is added into the proposal. Then, in order to improve the computational efficiency, we incorporate this gaussian process based proposal into KLD-Sampling particle filter to adapt the particle number over time by KL divergence. The proposed algorithm could use high-quality particles with necessary number to improve the estimation performance. The experiment results show its validity. Future work will firstly implement the proposed method for indoor mobile robot localization after learning the optimal proposal by data collection from map, then apply nonparametric techniques to handle the dual estimation for nonlinear high level control tasks in robotics.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.*, Cambridge, MA: The MIT Press, 2005.

[2] Stuart J.Russell and Peter .Norvig, *Artificial Intelligence: A Modern Approach (Thrid Edition)*, 2010.

[3] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Transactions of the ASME-Journal of basic Engineering*, 82(Series D), 1960, pp 35-45.

[4] A. Jazwinski, *Stochastic processes and filtering theory*, 1970.

[5] Simon J. Julier, Jeffrey K. Uhlmann and Hugh F. Durrant-Whyte, "A New Approach for Filtering Nonlinear Systems", *Proceedings of the American Control Conference*, vol. 3, 1995, pp 1628-1632.

[6] Jun S.Liu, Rong Chen, "Sequential Monte Carlo Methods for Dynamic Systems", *Journal of the American Statistical Association*, vol. 93, 1998, pp 1032-1044.

[7] Arnaud Doucet, Simon Godsill and Christophe Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statistics and Computing*, vol. 10, 2000, pp 197-208.

[8] Olivier Cappe, Simon J. Godsill and Eric Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo", *PROCEEDINGS OF THE IEEE*, vol. 95, 2007, pp 899-924.

[9] Zhe Chen, "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond", http://www.cse.buffalo.edu/~peter/refs/DataAssimilation/GenRefsTutorial\%s/Chen_2003.pdf, 2003.

[10] Fred Daum, and Jim Huang, "Particle Degeneracy: Root Cause and Solution", *Signal Processing, Sensor Fusion, and Target Recognition XX*, USA, 2011.

[11] Rudolph van der Merwe, Arnaud Doucet, Nando de Freitas and Eric Wan, "The Unscented Particle Filter", Technical Report, Engineering Department, Cambridge University, 2000.

[12] Christian Plagemann, Dieter Fox and Wolfram Burgard, "Efficient Failure Detection on Mobile Robots Using Particle Filters with Gaussian Process Proposals", *PROCEEDINGS OF IJCAI*, 2007, pp 2185-2190.

[13] Dieter Fox, "KLD-Sampling: Adaptive Particle Filters", *Neural Information Processing Systems*, 2001, pp 713-720.

[14] Jonathan Ko and Dieter Fox, "GP-BayesFilters: Bayesian filtering using Gaussian Processes prediction and observation models", *Auton Robot*, vol. 27, 2009, pp 75-90.

[15] Jonathan Ko and Dieter Fox, "Learning GP-BayesFilters via Gaussian Processes latent variable models", *Auton Robot*, vol. 30, 2011, pp 3-23.

[16] Carl Edward Rasmussen, and Christopher K.I. Williams, *Gaussian Process for Machine Learning*, MIT Press, 2006.

[17] Christopher M.Bishop, *Pattern Recognition and Machine Learning*, Springer Science + Business Media, LLC, 2006.

[18] Niclas Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications*, Ph.D thesis, Linkoping University, 1999.

[19] Patrick Dallaire, Camille Besse and Brahim Chaib-draa, "An approximation inference with Gaussian process to latent functinsnfrom uncertain data", *Neurocomputating*, vol. 74, 2011, pp 1945-1955.