Stochastic Optimal Control Using Gaussian Process Regression over Probability Distributions

Jana Mayer[1], Maxim Dolgov[2], Tobias Stickling[1], Selim Özgen[1], Florian Rosenthal[1], and Uwe D. Hanebeck[1]

***Abstract*— In this paper, we address optimal control of nonlinear stochastic systems under motion and measurement uncertainty with finite control input and measurement spaces. Such problems can be formalized as partially-observable Markov decision processes where the goal is to find policies via dynamic programming that map the information available to the controller to control inputs while optimizing a performance criterion. However, they suffer from intractability in scenarios with continuous state spaces and partial observability which makes approximations necessary. Point-based value iteration methods are a class of global approximate methods that regress the value function given the values at a set of reference points. In this paper, we present a novel point-based value iteration approach for continuous state spaces that uses Gaussian processes defined over probability distribution for the regression. The main advantages of the proposed approach is that it is nonparametric and therefore approximation quality can be adjusted by choosing the number and the position of reference points in the space of probability distributions. In addition, it provides a notion of approximation quality in terms of variance.**

I. INTRODUCTION

Real-world control problems such as robot navigation are generally affected by disturbances that often can be modeled as a stochastic process. In the case of robot navigation, uncertainties can arise, e.g., from external sources such as roughness of the underground or slip of the wheels. But also uncertainties in the theoretical assumptions such as imprecise dynamic models or incomplete information of the surroundings can have an impact. In addition, observations of the robot's state and its environment are usually available in form of measurements that are also subject to disturbances and only provide partial observability of relevant quantities. Stochastic optimal control and partially-observable Markov decision processes (POMDPs), respectively, are common frameworks to address such problems.

Both approaches determine policies that map the information available at a certain time step to control inputs by optimizing a performance criterion. In the stochastic optimal control framework, a cost function is minimized that not only takes into account the cost at the current time step but also future costs. Please note that in the POMDP framework usually a value function is maximized in an analogous manner. In this paper, we will refer to the term *value function* without loss of generality.

A solution to a stochastic control problem can be found using the dynamic programming (DP) algorithm that is based on Bellman's principle of optimality [1]. This algorithm relies on two key principles. First, it assumes the existence of sufficient statistics in form of a probability distribution that encompasses all the information up to the moment of policy evaluation. And second, it computes the policy by starting at the end of the planning horizon and going backwards in time to the current time step and at every time step choosing a control input that minimize the cumulative costs maintained in form of the value function from this time step to the end of the optimization horizon.

In general, DP for POMDPs is computationally intractable but for very few cases, e.g., if the state, control, and measurement spaces are finite and small [2], [3]. Under the assumptions of linear system and measurement equation, a quadratic cost function with deterministic parameters, and independent and identically distributed Gaussian system and measurement noises, DP can also be solved. This class of problems is referred as linear quadratic Gaussian control (LQG) [4]. For more general scenarios, approximate DP approaches are required.

The approaches available in literature can be classified into local and global methods. Local approaches operate on a single trajectory. First, an initial policy is chosen and executed. Then, going backward in time from the end of the optimization horizon, the policy is improved using local approximations of the value function. This procedure is repeated until a local optimum is found. Therefore this method is also known as trajectory optimization. For nonlinear control problems, a popular method is to extend the LQG using a second-order Taylor expansion of the value function and a linearization of the system equation [5]. An extension to partially-observable systems is given in [6]. The authors use an extended Kalman filter (EKF) as state estimator assuming measurement and process noise to be Gaussian.

Global value function approximations on the other hand use interpolation methods to determine the expected future costs at a state estimate from values available at a certain set of state estimates. Most approaches of this class are so called point-based value iteration methods. One important approach was given by Pineau et al. [7] where discrete state, measurement, and control spaces were assumed. An extension of this concept to problems with continuous state, control, and measurements spaces was done by Porta et al. [8], where the so-called α-vectors that represent the value function in discrete problems

*This work was supported by the German Research Foundation (DFG) under grant HA 3789/17-1.

[1]The authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. E-Mail: jana.mayer@kit.edu, tobias.stickling@student.kit.edu, selim.oezgen@kit.edu, florian.rosenthal@kit.edu, uwe.hanebeck@ieee.org

[2]Maxim Dolgov is with the Robert Bosch GmbH, Corporate Research. E-Mail: maxim.dolgov@de.bosch.com

were transformed into α-functions. As foundation, the authors showed that the value function over continuous state space is convex in the state estimates and piecewise-linear convex for discrete control inputs and measurements. In this approach, the measurement, system, and reward models were described by Gaussian mixtures and the state estimates by Gaussian mixtures or particle sets. Another important contribution to POMDPs for continuous state space was presented by Thrun [9]. In his Monte Carlo POMDP algorithm, the state estimates are represented by sets of particles. The value function is learned using a nearest neighbor method based on the Kullback-Leibler (KL) divergence. In [10] a finite-state controller is used to represent the policy. The policy is determined by sampling the state space and applying Monte Carlo simulations. This approach was extended in [11] by employing macro-actions and in [12] by pruning and optimizing the policy graph. Another extension is to utilize inductive learning in order to learn an effective discrete representation of the continuous state space parallel to the evaluation of the control problem [13]. In the field of Gaussian mixture continuous POMDPs, Burks et al. [14], [15] presented an approach for efficient hybrid continuous–discrete probabilistic observation likelihood models using softmax models. Besides, they also proposed a fast and scalable clustering-based Gaussian mixture condensation technique for large mixtures.

In this paper, we present a new point-based value iteration method that estimates the value function using Gaussian processes for regression. As we consider a stochastic control problem with imperfect state information, we work with probability distributions instead of states. To this end, a Gaussian process approach is required that can operate on inputs in form of probability distributions. For this purpose, we will use the approach from [16], a short description is given in section III. In the evaluation, we will compare the results of our approach with the results from Porta et al. [8].

II. PROBLEM FORMULATION

In this paper, we consider a finite-horizon stochastic control problem over a continuous state space and finite sets of control inputs $\mathcal{U} \in \mathbb{R}^{n_u}$ and measurements $\mathcal{Y} \in \mathbb{R}^{n_y}$. The discrete-time stochastic system may be described by the following dynamics

$$\begin{aligned} \underline{\boldsymbol{x}}_{k+1} &= \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) + \underline{\boldsymbol{w}}_k, \quad \underline{\boldsymbol{w}}_k \sim f_k^w, \\ \underline{\boldsymbol{y}}_{k+1} &= \underline{h}_{k+1}(\underline{\boldsymbol{x}}_{k+1}, \underline{u}_k) + \underline{\boldsymbol{v}}_{k+1}, \quad \underline{\boldsymbol{v}}_k \sim f_k^v \ . \end{aligned} \tag{1}$$

Affected by the control input $\underline{u}_k \in \mathcal{U}$ the system state $\underline{\boldsymbol{x}}_k \in \mathbb{R}^{n_x}$ changes according to the time-variant nonlinear function $\underline{a}_k$ and the system noise $\underline{\boldsymbol{w}}_k \in \mathbb{R}^{n_x}$ with probability distribution f_k^w. The feedback to the controller is given in form of measurements $\underline{\boldsymbol{y}}_k \in \mathcal{Y}$ that are related to the state $\underline{x}_k$ via the time-variant nonlinear measurement function $\underline{h}_k$ and the measurement noise $\underline{\boldsymbol{v}}_k \in \mathbb{R}^{n_y}$ distributed according to f_k^v. We assume independent and identically distributed noise processes. Furthermore, drawing samples from the distributions should be possible.

Given the dynamics (1), the goal consists in finding a function or *policy* that maps the information available to the controller while minimizing the cumulative cost function

$$\mathcal{J} = E\left\{ c_K(\underline{\boldsymbol{x}}_K) + \sum_{k=0}^{K-1} c_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) \middle| \mathcal{I}_k \right\}, \tag{2}$$

where c_k gives the costs at time step k and c_K the costs at the end of the planning horizon $K \in \mathbb{N}$, respectively. The cost function (2) is conditioned on the information set $\mathcal{I}_0 = \{f_0^x(\underline{x})\}$ that contains the probability distribution of the initial state $\underline{x}_0$ and the model (1). At time step k, the information set available to the controller is given by

$$\mathcal{I}_k = \{\underline{y}_k, \underline{u}_{k-1}, \mathcal{I}_{k-1}\} \ .$$

The control problem (1), (2) can be solved via dynamic programming (DP) where an estimate of the future performance criterion is maintained in form of value functions [17]. To this end, it first has to be reformulated in terms of *sufficient statistics* that condense the measurements $\underline{y}_{1:k}$, control inputs $\underline{u}_{1:k-1}$ and the initial distribution $f_0^x(\underline{x})$ into an estimate of the state at time step k in form of the probability distribution $f_k^x(\underline{x})$. Under this reformulation, the policy computed using DP becomes a function of the state estimates, i.e., the goal is to find $\pi_k(f_k^x(\underline{x}))$. Then, the Bellman recursion of DP can be formulated according to

$$\begin{aligned} V_K^*(f_K^x(\underline{x})) &= \underset{\underline{x}_K \sim f_K^x}{E} \left\{ c_K(\underline{\boldsymbol{x}}_K) \middle| \mathcal{I}_K \right\} \\ V_k^*(f_k^x(\underline{x})) &= \min_{\underline{u}_k} \underset{\underline{x}_k \sim f_k^x}{E} \left\{ c_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) + \gamma V_{k+1}^*(f_{k+1}^x) \middle| \mathcal{I}_k \right\}, \end{aligned} \tag{3}$$

where V_K^* denotes the optimal value function at the end of the planning horizon and V_k^* the optimal value function at time step k, and $\gamma \in [0, 1]$ is a discount factor that reduces the influence of far future incidences on the costs. The policy can then be recovered via

$$\pi_k(f_k^x(\underline{x})) = \arg\min_{\underline{u}_k} \underset{\underline{x}_k \sim f_k^x}{E} \left\{ c_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) + \gamma V_{k+1}^*(f_{k+1}^x) \middle| \mathcal{I}_k \right\} \ .$$

For continuous state spaces, (3) yields

$$V_K^*(f_K^x) = \int c_k(\underline{x}_K) f_K^x(\underline{x}_K) d\underline{x}_K \ , \tag{4}$$

and

$$\begin{aligned} V_k^*(f_k^x) = \min_{\underline{u}_k \in U} \int \Big[& c_k(\underline{x}_k, \underline{u}_k) \\ & + \gamma \sum_{j=1}^{|\mathcal{Y}|} p(\underline{y}_{k+1,j} | \underline{x}_k, \underline{u}_k) V_{k+1}^*(f_{k+1,\underline{y}_j,\underline{u}}^x) \Big] f_k^x(\underline{x}_k) \, d\underline{x}_k \ . \end{aligned} \tag{5}$$

The state estimate at the time step $k+1$

$$f_{k+1,\underline{y},\underline{u}}^x = p(\underline{x}_{k+1} | \underline{u}_k, \underline{y}_{k+1})$$

is defined by Bayes' law

$$f_{k+1,\underline{y},\underline{u}}^x = \frac{p(\underline{y}_{k+1} | \underline{x}_{k+1}) \int p(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k) f_k^x(\underline{x}_k) d\underline{x}_k}{\iint p(\underline{y}_{k+1} | \underline{x}_{k+1}) p(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k) f_k^x(\underline{x}_k) d\underline{x}_k d\underline{x}_{k+1}} \tag{6}$$

and the probability of observing the measurement $\underline{y}_{k+1}$ according to the previous state x_k and executed control input u_k is given by

$$p(\underline{y}_{k+1}|\underline{x}_k, \underline{u}_k) = \int p(\underline{y}_{k+1}|\underline{x}_{k+1}, \underline{u}_k) p(\underline{x}_{k+1}|\underline{x}_k, \underline{u}_k) d\underline{x}_{k+1}.$$

One method to solve (5) is point-based value iteration where the value function at each time step is only maintained for a finite set of reference points $\hat{f}^x_{0:k}$, the *reference state estimates*. To compute the values at the reference points $\hat{f}^x_k$ at time step k using Bellman recursion (3), we need to know the values at points f^x_{k+1} that result from $\hat{f}^x_k$ after applying $\underline{u}_k$. In general, the points f^x_{k+1} do not correspond to the reference points $\hat{f}^x_{k+1}$ for which the values are known and thus have to be determined by regression methods. As the state estimates are probability distributions, interpolating the exact value function can only be done by regression methods that can deal with inputs in form of probability distributions. In this work, we present a new approach using Gaussian processes over probability distributions to handle the regression problem in point-based value iteration.

III. PRELIMINARIES: GAUSSIAN PROCESSES OVER PROBABILITY DISTRIBUTIONS

An approach for Gaussian processes over probability distributions is given in [16]. In this work, distance measures for probability distributions are used to construct the covariance functions of the Gaussian process.

A. Gaussian Processes with Deterministic Inputs

First, we introduce Gaussian processes with deterministic inputs. A Gaussian process is a nonparametric regression method for nonlinear mappings

$$\boldsymbol{y} = g(\underline{x}) + \boldsymbol{w} \ ,$$

where a real-valued vector $\underline{x}$ is mapped to a scalar $\boldsymbol{y}$. The model imperfections and other disturbances are modeled with an independent zero-mean normal noise $\boldsymbol{w}$ with variance σ^2. A Gaussian process models the output y with a Gaussian distribution with mean $m(\underline{x})$ and covariance $C(\underline{x})$ that are both functions of $\underline{x}$. Thus, we can write

$$y \sim \mathcal{GP}(m(\underline{x}), C(\underline{x})) \ .$$

Given a training dataset $D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_N, y_N)\}$, we can estimate a Gaussian process and then use it to regress the output g_* at an input vector $\underline{x}_*$. To this end, we first construct the sample mean vector $\underline{\mu}$ and the sample covariance Σ according to

$$\underline{\mu} = \begin{bmatrix} \mu(\underline{x}_1)^\top & \mu(\underline{x}_2)^\top & \ldots & \mu(\underline{x}_N)^\top \end{bmatrix}^\top \ ,$$

$$\Sigma = \begin{bmatrix} \kappa(\underline{x}_1, \underline{x}_1) & \ldots & \kappa(\underline{x}_1, \underline{x}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\underline{x}_N, \underline{x}_1) & \ldots & \kappa(\underline{x}_N, \underline{x}_N) \end{bmatrix} \ ,$$

where $\mu(\underline{x})$ denotes the mean function and $\kappa(\underline{x}, \underline{x}')$ the covariance function. Then, the hyperparameters of the Gaussian process, i.e., the parameters of the mean and the covariance functions, are determined by maximizing either the posterior or the likelihood of the training outputs at the training inputs. Finally, the mean m_* and the variance σ^2_* of the prediction g_* at $\underline{x}_*$ can be calculated via

$$m_* = \mu(\underline{x}_*) + K^\top \left[\Sigma + \sigma^2 I\right]^{-1} \left(\underline{y} - \underline{\mu}\right) \ ,$$
$$\sigma^2_* = \kappa(\underline{x}_*, \underline{x}_*) - K^\top \left[\Sigma + \sigma^2 I\right]^{-1} K \ ,$$

where $K^\top = \begin{bmatrix} \kappa(\underline{x}_*, \underline{x}_1) & \kappa(\underline{x}_*, \underline{x}_2) & \ldots & \kappa(\underline{x}_*, \underline{x}_N) \end{bmatrix}$.

An important advantage of Gaussian processes compared to many other regression methods is that they provide the variance σ^2_* of the regressed output g_* that reflects the estimation quality. Furthermore, Gaussian processes usually perform better compared to parametric methods such as neural networks in problems where little training data is available because they tend to overfit less. A more detailed introduction to Gaussian processes is available in [18].

B. Gaussian Processes over Probability Distributions

While classical Gaussian process formalism only allows for real-valued vectors as inputs, our application requires Gaussian processes that can regress a real scalar function over probability distributions. To this end, we will use the framework presented in [16]. The main idea of this framework is to employ covariance functions that are based on distances between probability distributions. To visualize this concept, consider the stationary squared exponential covariance function

$$\kappa(\underline{x}_i, \underline{x}_j) = \alpha^2 \exp\left(-\frac{1}{2} \frac{(\underline{x}_i - \underline{x}_j)^T (\underline{x}_i - \underline{x}_j)}{l^2}\right)$$

for deterministic vector-valued inputs. This function is referred to as *stationary* because it only depends on the squared Euclidean distance $(\underline{x}_i - \underline{x}_j)^T (\underline{x}_i - \underline{x}_j)$ between the inputs. The main notion of [16] is to substitute the Euclidean distance for a distance measure between probability distributions $d(f^x_i, f^x_j)$, which yields the new covariance function

$$\kappa(f^x_i, f^x_j) = \alpha^2 \exp\left(-\frac{1}{2} \frac{d(f^x_i, f^x_j)^2}{l^2}\right) \ . \tag{7}$$

Other covariance functions that depend on the distance between the inputs can be extended in the same way. Furthermore, a combination of such stationary covariance functions with non-stationary ones can be done as in the classical Gaussian process formalism. Finally, the hyperparameter estimation of the Gaussian process and the prediction remain also the same.

Different distance measures exist in literature [19] that can be used in the described framework. In this paper, we will refer to the modified Cramér – von Mises distance (mCvMd) [20] because it allows for computation of distances between distributions given in form of Dirac mixtures [21].

IV. PROPOSED APPROACH

As mentioned in the introduction, we address the considered control problem using a point-based value iteration approach where the value is maintained only at a set of reference state estimates $\hat{f}_k^x$ that are given in form of probability distributions. The main idea of our approach is to approximate the value function $V_{k+1}^*(f_{k+1,\underline{y}_j}^x)$ in (5) that is required to determine the value function at a time step k using Gaussian processes over probability distributions. Details of the implementation are given in Algorithm 1.

Algorithm 1 Stochastic Optimal Control with Gaussian Processes

1: $k = K$
2: Generate set of state estimates $\hat{f}_{i,k}^x$, $i = 1, ..., N$
3: Initialize value function $V_K^*(\hat{f}_{i,k}^x)$ ▷ (8)
4: **while** $k > 1$ **do**
5: New training data $\left\{\hat{f}_{1,k}^x, ..., \hat{f}_{N,k}^x\right\}, \left\{V_k^*(\hat{f}_{1,k}^x), ..., V_k^*(\hat{f}_{N,k}^x)\right\}$ for GP
6: Estimate hyperparameters of GP
7: $k = k - 1$
8: **for** $i = 1, ..., N$ **do**
9: Filter - prediction $\rightarrow f_{i,k+1,\underline{u}}^p$
10: Filter - update $\rightarrow f_{i,k+1,\underline{y},\underline{u}}^x$
11: Apply GP on test data $f_{i,k+1,\underline{y},\underline{u}}^x$ and
12: get $V_{k+1}^*(f_{i,k+1,\underline{y},\underline{u}}^x)$
13: Calculate $V_k^*(\hat{f}_{i,k}^x) \rightarrow \pi_k(\hat{f}_{i,k}^x)$ ▷ (9)
14: **end for**
15: **end while**

In order to perform Bellman recursion (3), we need to find a set of suitable reference state estimates in form of probability distributions that sufficiently cover the relevant state space. For this purpose, we use the Monte-Carlo approach of Porta et al. [8] where random control inputs from the control input set are applied on the state estimate alternating with filter steps using random admissible measurements from the measurement set. To get an optimal coverage of the considered continuous space by a predetermined number of state estimates N, we use the modified Cramér – von Mises distance (mCvMd), introduced in section III-B, keeping only state estimates with a predetermined minimum distance. We utilize the same set of reference state estimates $\hat{f}_{1,K}^x, ..., \hat{f}_{N,K}^x$ in every time step, but for better understanding we will keep the index k in the equations.

The value function $V_k^*(\hat{f}_{i,k}^x)$ at each time step k depends on the value function $V_{k+1}^*(f_{i,k+1,\underline{y},\underline{u}}^x)$ of the future time step $k+1$ (5). As mentioned in Sec. II, $f_{i,k+1,\underline{y},\underline{u}}^x$ is the posterior distribution of the prior reference state estimate $\hat{f}_{k,i}^x$, applying control input $\underline{u}_k$ and measurement $\underline{y}_k$. It can be easily seen that the state estimates at time step k in general do not correspond to the state estimates at time step $k+1$ as illustrated in Fig. 1. To overcome this issue, we regress the value function at the posterior state estimates $V_{k+1}^*(f_{i,k+1,\underline{y},\underline{u}}^x)$ using Gaussian processes over probability

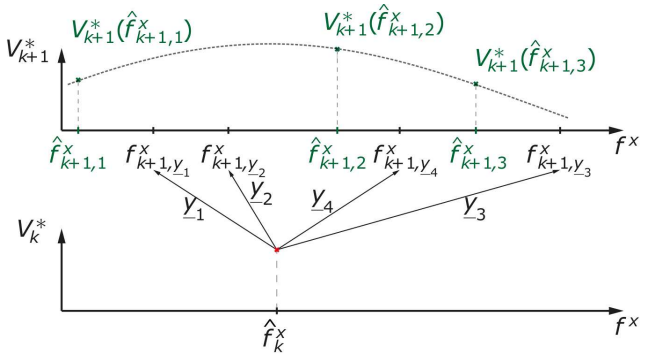


Fig. 1: Evolution of the state estimates from one time step to the next applying one action and several measurements.

distributions, introduced in section III-B. The values of the reference state estimates calculated at the previous recursion step (5) are used as training data for the Gaussian processes. This is reasonable because they describe the system at the same time step as the values of the posterior distributions $V_{k+1}^*(f_{i,k+1,\underline{y},\underline{u}}^x)$ of the current recursion step.

So far the state estimates were only specified as arbitrary probability distributions. For the implementation, we have to utilize a tractable representation of the probability distributions. For reasons of clarity, we will omit the index i for the state estimate in the following section. We choose Dirac mixtures to represent the state estimates, i.e., a set of weighted samples

$$f_k^x(\underline{x}) = \sum_{m=1}^{M} w_{k,m} \delta(\underline{x}_k - \underline{x}_{k,m}) \ ,$$

where $\underline{x}_{k,m}$ are the sample positions and $0 < w_{k,m} \leq 1$ the corresponding weights with $\sum_{m=1}^{M} w_{k,m} = 1$. Dirac mixtures can be used as approximations for arbitrary probability distributions. Alternative representations include Gaussians or Gaussian mixtures.

A nonlinear estimator, working with state estimates in the form of Dirac mixtures, is needed to perform the state estimation (6). We use the sampling importance resampling (SIR) particle filter [22], but in general every particle filter can be used.

Applying Dirac mixtures to the value function (4) and (5) yields

$$V_K^*(\hat{f}_K^x) = \sum_{m=1}^{M} w_{K,m} c_K(\underline{x}_{K,m}) \tag{8}$$

and

$$V_k^*(\hat{f}_k^x) = \min_{\underline{u}_k \in U} \Big[\sum_{l=1}^{L} w_{k,l} c_k\left(\underline{x}_{k,l}, \underline{u}_k\right) + \gamma \sum_{j=1}^{|Y|} \sum_{m=1}^{M} w_{k+1,m} p(\underline{y}_{k+1,j} | \underline{x}_{k+1,m}, \underline{u}_k) V_{k+1}^*(f_{k+1,\underline{y},\underline{u}}^x) \Big] \ , \tag{9}$$

where the state estimates at the current time step k are approximated by

$$\hat{f}_k^x = \sum_{l=1}^{L} w_{k,l}\delta(\underline{x}_k - \underline{x}_{k,l}) \ .$$

The state estimates at the prediction step of the particle filter after executing control input $\underline{u}_k$ are

$$f_{k+1}^p = \sum_{m=1}^{M} w_{k+1,m}\delta(\underline{x}_{k+1} - \underline{x}_{k+1,m}) \ .$$

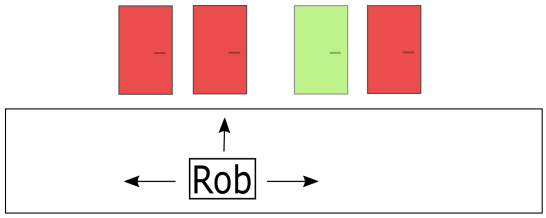
As before, the state estimates after the update step of the particle filter are referred to as $f_{k+1,\underline{y},\underline{u}}^x$.

We determine $V_{k+1}^*(f_{k+1,\underline{y},\underline{u}}^x)$ using Gaussian processes over probability distributions. The covariance function of the Gaussian process depends on the distance between the probability distributions, see section III-B.

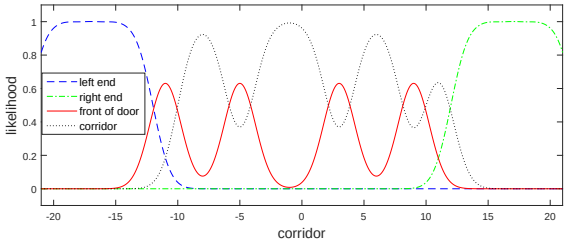
By executing Algorithm 1, the policy $\pi_K(\hat{f}_{i,k}^x)$ for horizon K is determined. Now the agent can act according to the policy. The information about the agent's initial position is given to the agent in form of a initial state estimate, e.g., if the position is unknown a uniform distribution will be assumed. To reach its goal the agent will execute actions according to the policy. As the policy maps the state estimates to actions, the agent needs to determine his current state estimate in every time step. Knowing the initial state estimate, the executed actions and the measurements taken in every time step the current position of the agent can be estimated using a particle filter. As mentioned before, the state estimates after executing the filter are in general not corresponding to the state estimates before. As we determined the policy only for a fixed set of state estimates, we have to find the corresponding control input for the calculated state estimate at every time step. Therefore we determine the nearest reference state estimate according to the mCvMd and execute the control input that is mapped by the policy on this reference state estimate.

V. EVALUATION

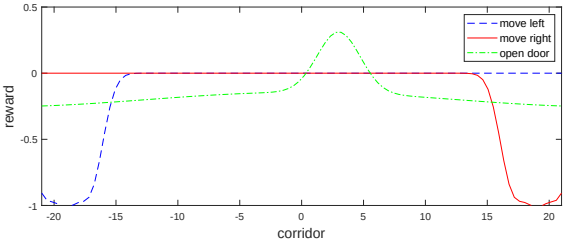
For the evaluation we implemented the corridor problem from Porta et al. [8]. The properties of the experiment are shown in Fig. 2. In this problem, a robot acts in a one-dimensional corridor and its mission is to open one particular door out of four doors, while its position is only known uncertainly. The target door is the second from the right in Fig. 2a. In our algorithm the state estimate of the robot is given in form of Dirac mixtures. In the approach of Porta et al. [8] the state estimate is approximated as well by Dirac mixtures or alternatively by Gaussian mixtures. The robot can execute three different control inputs: "move left", "move right", and "open door". In Fig. 2b, the measurement model is shown. In contrast to the assumptions in the problem formulation, where the likelihood is derived from the measurement equation (1), here the likelihood is given directly by Gaussian mixtures. Furthermore, in this experiment the measurements are assumed to be independent of the control inputs, which means the measurement function $\underline{h}_{k+1}$ in (1) is



(a) Setup of corridor problem.



(b) Measurement model.

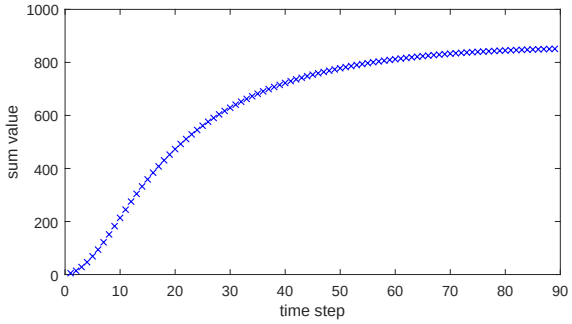


(c) Reward model.

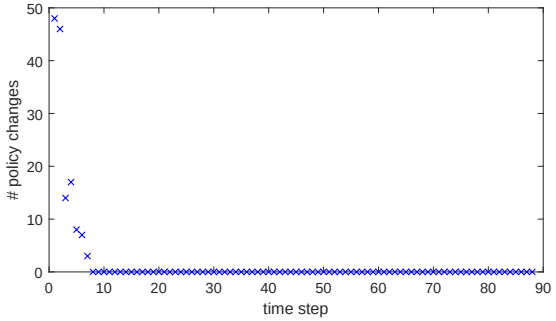
Fig. 2: Properties of the corridor problem invented by Porta et al. [8].

independent of $\underline{u}_k$. As this problem was invented in a POMDP framework, maximum rewards instead of minimum costs are aimed. The reward model is also modeled by Gaussian mixtures for every control input, see Fig. 2c. Our algorithm is working optimally only for small system noise, therefore we consider zero-mean Gaussian noise with covariance 10^{-5}. All other parameters are adopted from the online available implementation of Porta et al. [8]. We consider 200 reference state estimates each represented by a Dirac mixture with 40 samples. Their evolution, applying control inputs and measurements, is calculated using the SIR particle filter of the Nonlinear Estimation Toolbox [23]. In the Gaussian process, we use the squared exponential covariance function (7).

In the next section we first give a proof-of-concept for our approach solving the corridor problem, then we compare our results to the results of Porta et al. [8]. The corridor problem has a discount factor of 0.95, thus the value function is supposed to converge after a finite number of time steps. In Fig. 3a, the sum of the values of all reference state estimates $\sum_{i=1}^{N} V_k^*(\hat{f}_i^x)$ for every time step is shown. It can be seen, that the value function converges after approximately 80 time steps. The policy converges even faster after approximately eight time steps, which is presented in Fig. 3b.



(a) Sum over values of all state estimates.



(b) Number of changes in the policy per time step.

Fig. 3: Evolution of the value function and the number of policy changes for horizon K = 90.

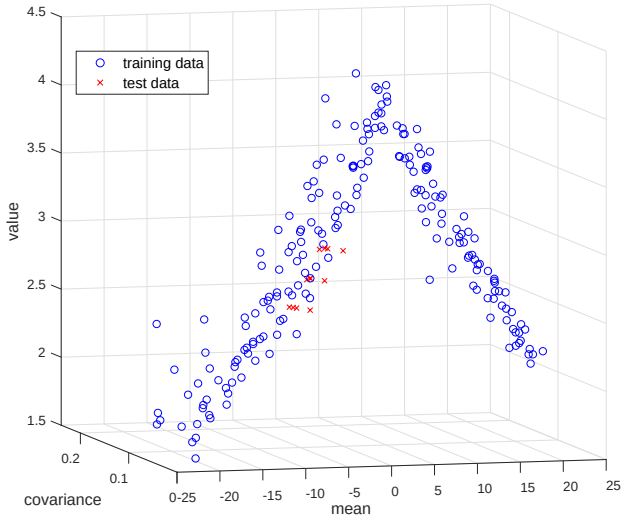


Fig. 4: Values over mean and covariance of the state estimates. The blue circled are the training data and the red crosses are the values determined at the test data using Gaussian processes over probability distributions.

The regression of the value function using Gaussian processes is illustrated in Fig. 4. As we mentioned in section IV, the training data are the values at the reference state estimates calculated at the previous recursion step (blue circles). For the current recursion step k, we need the values of the state estimates at time step $k+1$, see equation (9). Those state estimates are determined by applying all combinations of actions and measurements on the reference state estimates and act as test data in the Gaussian process. In Fig. 4, we show the regressed values for those state estimates belonging to one reference state estimate at the current recursion step k (red crosses). Please note, the state estimates are given in form of Dirac mixtures. In the Gaussian process, for every input in form of a Dirac mixture a output in form of a scalar value is determined. For simplicity, the value function is drawn over the means and covariances of the Dirac mixtures in Fig. 4.

Finally, we compare the results of our approach with the results of Porta et al. [8]. Their algorithm solves the control problem for an infinite horizon. For this, they determine the policy by approximating the value function iteratively. As shown before, the policy determined by our algorithm as well gets stationary after some time steps. For this reason the results after the first time steps may not be comparable, but after the policy converged, they are.

For the simulation of the robot walk, we assume contrary to the Porta et al. [8] corridor problem, that the starting point of the robot is known with a small uncertainty, here the initial state estimate is a Gaussian distribution with covariance 0.2. In our algorithm, we convert the Gaussian distribution into a Dirac mixture. Therefore we use the *GaussianSamplingLCD* of the Nonlinear Estimation Toolbox [23], which is a Gaussian sampling technique based on the localized cumulative distribution (LCD). In Fig. 5, we show the results for a start state estimate with mean -2 assuming a 30 time steps robot walk. We determined the policy for different run times executing our algorithm ten times. Afterwards we simulated the robot walk 200 times per policy. The results of Porta et al. [8] were calculated running the solver ten times and executing the simulation 20 times per ascertained policy at different run times. As they presented two different approaches for state estimates in form of Gaussian mixtures and Dirac mixtures, we determined the results for both. The simulation was implemented in MATLAB R2018a and run on a Intel Core i5-3570 at 3.40 GHz under Windows 10.

Even though our approach is intended for finite horizon problems, we achieve better results for small run times than Porta et al. [8]. After a solver run time of approximate 300 s, our results are equal to the results of Porta et al. [8] using Gaussian mixtures to represent the state estimates. Comparing the approaches that use states estimates in form of Dirac mixtures which allows a representation of arbitrary distributions, we outperform the approach of Porta et al. [8].

VI. CONCLUSIONS

In this paper, we introduced a novel point-based value iteration algorithm for problems with continuous state spaces and finite control and measurement spaces. Due to partial observability, the problem is addressed in the space of probability distributions that constitute sufficient statistics

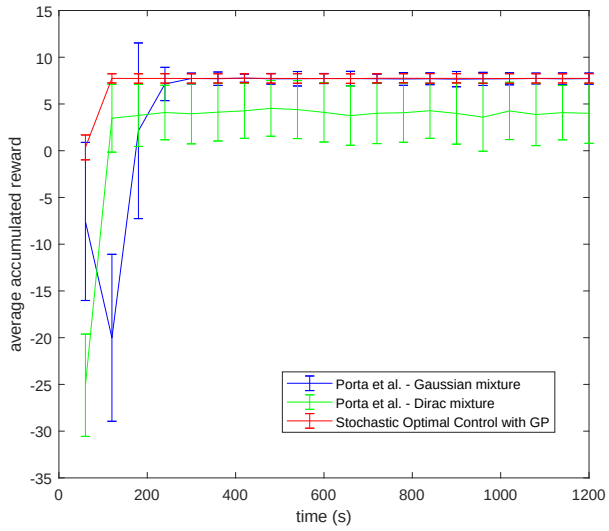


Fig. 5: Results of a 30 time steps robot walk simulation using policies determined after different solver run times.

obtained via filtering. In order to regress the value function during Bellman recursion, we use Gaussian processes that are defined over probability distributions. The presented approximation method does not pose restrictive assumptions onto the class of probability distributions. In our experiments, the proposed algorithm showed competitive performance to state-of-the-art approaches, in the case that the starting point is known with a small uncertainty.

For more complex value functions more reference state estimates might be needed. In this case, not only the Bellman update have to be executed more often in every recursion step, but also the number of training points in the Gaussian process increase. As the Gaussian process has a complexity of $\mathcal{O}(N^3)$, using scalable Gaussian processes [24] might be necessary.

In future work, we intend to improve the algorithm by (1) actively placing reference points for value function approximation, (2) performing a more sophisticated value update that, for example, considers attraction of certain state space areas and updates these areas more often, and (3) consideration of value approximation quality that is available in form of variance during Bellman recursion. This improvements should lead to a sufficient performance also in case of larger system noise. In addition, we plan to extend our approach to continuous spaces of control inputs and measurements.

REFERENCES

[1] R. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory*. Academic Press, 1965.

[2] R. Smallwod and E. J. Sondik, "The Optimal Control of Partially Observable Markov Processes over a Finite Horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[3] E. J. Sondik, "The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs," *Operations Research*, vol. 26, no. 2, pp. 282–304, 1978.

[4] M. Athans, "The Role and Use of the Stochastic Linear Quadratic Gaussian Problem in Control System Design," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.

[5] E. Todorov and W. Li, "A Generalized Iterative LQG Method for Locally-optimal Feedback Control of Constrained Nonlinear Stochastic Systems," in *Proceedings of the 2005 American Control Conference (ACC 2005)*, 2005.

[6] W. Li and E. Todorov, "Iterative Linearization Methods for Approximately Optimal Control and Estimation of Non-linear Stochastic System," *International Journal of Control*, vol. 80, no. 9, pp. 1439–1453, 2007.

[7] J. Pineau, G. Gordon and S. Thrun, "Point-based Value Iteration: An Anytime Algorithm for POMDPs," in *Proceedings of the International Joint Conference on Artificia Intelligence*, 2003.

[8] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-Based Value Iteration for Continuous POMDPs," *Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.

[9] S. Thrun, "Monte Carlo POMDPs," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1064–1070, 2000.

[10] H. Bai, D. Hsu, W.S. Lee, and V.A. Ngo, "Monte Carlo Value Iteration for Continuous-state POMDPs," in *in Proceedings of the Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010.

[11] Z. W. Lim, D. Hsu, W. S. Lee, "Monte Carlo Value Iteration with Macro-Actions," *Advances in Neural Information Processing Systems (NIPS)*, vol. 24, 2011.

[12] W. Qian, Q. Liu, Z. Zhang, Z. Pan and S. Zhong, "Policy Graph Pruning and Optimization in Monte Carlo Value Iteration for Continuous-state POMDPs," in *in Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016.

[13] S. Brechtel, T. Gindele, and R. Dillmann, "Solving Continuous POMDPs: Value Iteration with Incremental Learning of an Efficient Space Representation," in *Proceedings of the International Conference on Machine Learning (ICML 2013)*, 2013.

[14] L. Burks and N. Ahmed, "Optimal Continuous State POMDP Planning with Semantic Observations," in *in Proceedings of the IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.

[15] L. Burks, I. Loefgren and N. Ahmed, "Optimal Continuous State POMDP Planning with Semantic Observations: A Variational Approach," *arXiv:1807.08229*, 2018. [Online]. Available: https://arxiv.org/abs/1807.08229

[16] M. Dolgov and U. D. Hanebeck, "A Distance-based Framework for Gaussian Processes over Probability Distributions," *arXiv preprint arXiv:1809.09193*, 2018. [Online]. Available: https://arxiv.org/abs/1809.09193

[17] D. P. Bertsekas, "Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC," *European Journal of Control*, vol. 11, no. 4-5, pp. 310–334, 2005.

[18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[19] V. M. Zolotarev, "Probability Metrics (in Russian)," *Teoriya Veroyatnostei i ee Primeneniya*, vol. 28, no. 1, pp. 278–302, 1983.

[20] U. D. Hanebeck and V. Klumpp, "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance," in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008.

[21] U. D. Hanebeck, "Optimal Reduction of Multivariate Dirac Mixture Densities," *Automatisierungstechnik*, vol. 63, no. 4, 2015.

[22] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[23] J. Steinbring, "The Nonlinear Estimation Toolbox," 2017. [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[24] H. Liu, Y. Ong, X. Shen and J. Cai,, "When Gaussian Process Meets Big Data: A Review of Scalable GPs," *arXiv:1807.01065*, 2018. [Online]. Available: https://arxiv.org/abs/1807.01065