# Stochastic Optimal Control using Local Sample-based Value Function Approximation

Maxim Dolgov[1], Gerhard Kurz[2], Daniela Grimm[2], Florian Rosenthal[2], and Uwe D. Hanebeck[2]

***Abstract*— In stochastic optimal control and partially-observable Markov decision processes, trajectory optimization methods iteratively deform a reference trajectory in a space of probability distributions such that the performance criterion associated with the problem attains an optimum. Related state-of-the-art trajectory optimization approaches are restricted to the space of Gaussian probability distributions where during optimization they perform second-order Taylor expansion of the value function at the parameters of the Gaussian, i.e. the mean and the covariance. In this paper, we propose a novel approach where trajectory optimization is performed in the space of Dirac distributions and the Taylor expansion of the value function is done at the positions of its samples. By doing so, we are able to deal with non-Gaussian distributions because Dirac distributions are often used to approximate arbitrary probability distributions. The proposed approach is demonstrated in a simulation.**

## I. Introduction

Imagine a scenario where a robot has to navigate through an environment using its own dynamical model, a map of the environment, and sensor measurements as feedback. The motion of the robot has to be designed such that a performance criterion defined in terms of costs gets minimized. Alternatively, the performance criterion can be defined in terms of a value that has to be maximized. In the described scenario, uncertainty can arise from multiple sources. First, the parameters of the robot's dynamic model may be imprecise or the map of the environment may be incomplete. Second, external forces may affect the robot's movement. And finally, the measurements may be disturbed by noise. If these uncertainties are statistically quantified, the described problem can be addressed within the framework of stochastic optimal control or partially-observable Markov decision processes (POMDPs), where we seek control policies that map the information available to the controller to control inputs [1].

The standard approach in stochastic optimal control consists in (1) redefining the problem in terms of the state estimate maintained as a probability distribution that condenses the information available to the controller, and (2) computing the policies using *Dynamic Programming* (DP). The main notion of the DP algorithm is to start at the end of the planning horizon and to iterate backwards while choosing the control policy such that the cumulative costs from the current step of the planning horizon to its end, the so-called costs-to-go, attain a minimum. By doing so, DP exploits Bellman's principle of optimality [2]. If the problem is well-posed, e.g., if the costs are discounted or the state distribution has a limit, DP even works for infinite prediction horizons.

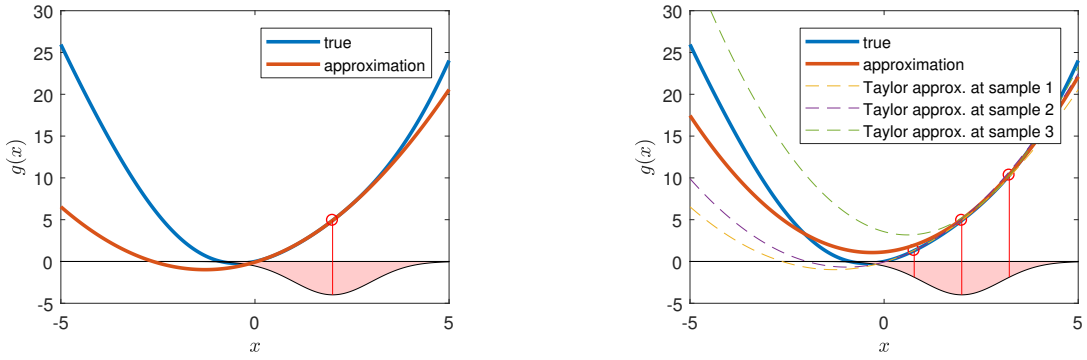
Unfortunately, DP is intractable unless the spaces of the states, the measurements, and the control inputs are finite and sufficiently small [3], or the system dynamics and the measurement equation are linear and have deterministic parameters, the noises that affect the system are independent and identically distributed with Gaussian probability distributions, and the cost function is quadratic and has deterministic parameters. The latter class of problems is referred to as Linear Quadratic Control (LQG) [4]. In other scenarios, DP is intractable because the separation between control and state estimation does not hold, i.e., the choice of current control inputs influences the quality of future state estimation and vice versa. This issue occurs because the latter influences the future decision making and therefore, the current costs-to-go [5]. Also, the optimal solution of DP requires that the costs-to-go are maintained for all possible state estimates, which is intractable because the space of general state estimates is infinite-dimensional. Of course there are other issues with DP, for example, the optimization problem associated with the choice of optimal control inputs is usually non-convex or the representation of state estimate is not parameterizable. Therefore, approximate approaches to DP are of interest.

The approximate DP approaches available in literature can be distinguished into *global* and *local* methods. Global approaches use function approximation methods in order to interpolate the costs-to-go using the values of the costs-to-go that are available for a set of state estimates. Methods that belong to this class of DP approaches are usually referred to as the Point-based Value Iteration methods [6], where the 'points' are probability distributions that represent the state estimates. A prominent global DP approach was presented by Thrun in [7], where the state estimates are represented using Dirac distributions, i.e., discrete probability distributions over a continuous domain. In order to interpolate the costs-to-go, Thrun uses a nearest-neighbor approach that relies on the Kullback-Leibler (KL) divergence [8] to determine the closeness of state estimates[1]. An important theoretical contribution to global DP approaches was presented by Porta et al. in [6],

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

[2]The authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. E-Mail: gerhard.kurz@kit.edu, florian.rosenthal@kit.edu, uwe.hanebeck@ieee.org

[1]Please note that the KL is not a distance measure because it does not satisfy the necessary axioms. Furthermore, it is not defined for Dirac distributions. Therefore, Thrun uses a Parzen-window smoothing method [9] to convert the Dirac distribution into a continuous distribution.




(a) Taylor approximation at the mean 2.

(b) Sample-based Taylor approximation at three locations.

Fig. 1: Approximation error of the second-order Taylor and sample-based second-order Taylor expansions of the function $a(x) = x^2 + \sin(x)$ for the Gaussian distribution $\mathcal{N}(2, 1)$.

where the authors generalize the results from [10] and [11] for problems with finite spaces of states, measurements, and control inputs to problems with a continuous state space. In this work, the authors show that the costs-to-go are piece-wise linear continuous in the state estimates if the spaces of measurements and control inputs, and also the planning horizon are finite. Furthermore, they extend the notion of $\alpha$-vectors that encode the optimal policy in stochastic optimal control problems with finite state spaces to the notion of $\alpha$-functions for problems with a continuous state space. The results from [6] are combined with a state aggregation heuristic in [12] and a policy representation using a finite-state controller in [13]. A global DP approach with Gaussian state estimates is presented in [14]. However, the parametrized state estimate representation limits the class of admissible problems.

In contrast to global DP approaches, local approaches perform DP along a reference trajectory of state estimates and control inputs. By doing so, the trajectory is iteratively deformed until the control performance criterion attains a local optimum. Therefore, local DP approaches are also referred to as *trajectory optimization* methods. The main advantage of local DP approaches is that it is not necessary to maintain the costs-to-go for the entire space of possible state estimates. However, it is not guaranteed that the global optimum will be found when using the vanilla version o this approach. A prominent branch of trajectory optimization methods relies on the extension of linear quadratic approaches to nonlinear control problems [15]. This methods are closely related to Differential Dynamic Programming [16]. An extension of the approach from [15] to the class of problems considered in this paper is presented in [17], where the authors use an Extended Kalman Filter (EKF) as the state estimator and perform second-order Taylor expansion of the costs-to-go at the mean of the Gaussian state estimate. We will refer to this approximation method as *EKF-based*. The parameters of the EKF and the control law that is affine in the mean of the state estimate are computed iteratively. In [18], van den Berg et al. also consider problems with Gaussian state estimates. In contrast to [17], they propose a closed-loop formulation of the control problem in terms of the Gaussian state estimate. By doing so, the Taylor expansion can be performed at the parameters of the Gaussians. Furthermore, van den Berg et al. use rapidly-exploring random trees [19] in order to generate a good initial reference trajectory, which increases the chance of finding the global optimum of the control problem.

In this paper, we extend the results from [17], [18] to sample-based representation of state estimates in terms of Dirac distributions. By doing so, we are able to address stochastic optimal control problems with non-Gaussian states. This representation of state estimates allows us to derive an approximation scheme for the costs-to-go that performs a second-order Taylor expansion at the positions of the samples of the Dirac distributions. We will refer to this scheme as *sample-based* or *statistical* Taylor approximation. In nonlinear filtering, sample-based approaches have shown to be superior to approximations based on Taylor series in terms of approximation quality and robustness [20], especially in problems with large noise covariances and strongly nonlinear dynamics and measurement functions. An example of this observation is depicted in Fig. 1, where the second-order Taylor series and the sample-based second-order Taylor approximations of $a(x) = \sin(x) + x^2$ for the Gaussian $\mathcal{N}(2, 1)$ are compared.

## II. Problem Formulation

In this section, we formulate the considered problem and introduce the basic concept of DP and some preliminaries. We consider the stochastic system with time-variant dynamics

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

where $\underline{\boldsymbol{x}}_k \in \mathbb{R}^{n_x}$ is the state of the system, $\underline{u}_k \in \mathbb{R}^{n_u}$ the control input that is generated by the controller at time step $k$, $\underline{\boldsymbol{w}}_k \in \mathbb{R}^{n_w}$ the process noise with distribution $p_k^w(\underline{w}_k)$, $\underline{\boldsymbol{y}}_k \in \mathbb{R}^{n_y}$ the measurement that is fed back to the controller at time step $k$, and $\underline{\boldsymbol{v}}_k \in \mathbb{R}^{n_v}$ the measurement noise with distribution $p_k^v(\underline{v}_k)$. For simplicity, we assume that the noises are independent and identically distributed, and that they can be sampled. The cost function associated with the considered control problem is given by

$$\mathcal{J} = \mathrm{E}\left\{ \mathcal{C}_K(\underline{\boldsymbol{x}}_K) + \sum\nolimits_{k=0}^{K-1} \mathcal{C}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) \right\} \ , \tag{2}$$

where $K \in \mathbb{N}$ denotes the length of the planning horizon and the functions $\mathcal{C}_K(\cdot)$ and $\mathcal{C}_k(\cdot,\cdot)$ return the costs incurred at time steps $k = 0, 1, \ldots, K$ by the state estimates $p_k^x(\underline{x}_k)$ and the corresponding control inputs $\underline{u}_k$. In our approach, we require that in the positive definite sense

$$\begin{gathered} \frac{\partial^2}{\partial \underline{x}_k^2} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) \geq 0 \ , \quad \frac{\partial^2}{\partial \underline{u}_k^2} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) > 0 \ , \\ \begin{bmatrix} \frac{\partial^2}{\partial \underline{x}_k^2} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) & \frac{\partial^2}{\partial \underline{x}_k \underline{u}_k} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) \\ \frac{\partial^2}{\partial \underline{u}_k \underline{x}_k} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) & \frac{\partial^2}{\partial \underline{u}_k^2} \mathcal{C}_k(\underline{x}_k, \underline{u}_k) \end{bmatrix} \geq 0 \ . \end{gathered} \tag{3}$$

The above assumption is necessary to produce well-defined local optimization problems during the iterative control law computation presented in this paper.

Instead of computing the individual control inputs, we seek a policy $\underline{\pi}_k(p_k^x(\underline{x}_k))$ for $k = 0, \ldots, K-1$ given an initial state estimate $p_0^x(\underline{x}_0)$ that during runtime maps the actual state estimates $p_k^x(\underline{x}_k)$ to control inputs such that the costs (2) attain a minimum. In this paper, we assume that the state estimates $p_k^x(\underline{x}_k)$ are maintained in form of Dirac distributions, i.e.,

$$p_k^x(\underline{x}_k) = \sum\nolimits_{i=1}^{N_x} \alpha_k^i \delta(\underline{x}_k - \underline{x}_k^i) \ , \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta, and $\underline{x}_k^i$ are the positions of the samples and $\alpha_k^i \in (0,1]$ their weights with $\sum_{i=1}^{N_x} \alpha_k^i = 1$. The estimate $p_k^x(\underline{x}_k)$ is obtained from the information set $\mathcal{I}_k = \{p_0^x(\underline{x}_0), \underline{\boldsymbol{y}}_1, \ldots, \underline{\boldsymbol{y}}_k, \underline{u}_0, \ldots, \underline{u}_{k-1}\}$ using an admissible filter such as the Unscented Kalman Filter (UKF) [21], the randomized UKF [22], or the particle filter [23]. As we will see later, the proposed statistical Taylor approximation induces the affine policy $\underline{\pi}_k(\underline{x}_k) = \mathbf{L}_k \mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \underline{d}_k$, whose parameters $\mathbf{L}_k$ and $\underline{d}_k$ can be computed using the presented algorithm.

As outlined in the introduction, the general approach to solve a stochastic optimal control problem is to apply DP. The main idea of this approach is to define the Bellman recursion

$$\begin{aligned} V_K(p_K^x(\underline{x}_K)) &= \mathrm{E}\{\mathcal{C}_K(\underline{\boldsymbol{x}}_K)\} \ , \\ V_k(p_k^x(\underline{x}_k)) &= \inf_{\underline{u}_k} \mathrm{E}\left\{ \mathcal{C}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) + V_{k+1}(p_{k+1}^p(\underline{x}_{k+1} | \underline{\boldsymbol{y}}_{k+1})) \right\} , \end{aligned} \tag{5}$$

where $V_K(\cdot)$ and $V_k(\cdot)$ are referred to as *value functions*. The probability distribution $p_{k+1}^p(\underline{x}_{k+1}|\underline{y}_{k+1})$ for a particular measurement $\underline{y}_{k+1}$ can be computed using the Bayes' law according to

$$\begin{aligned} p_{k+1}^p(\underline{x}_{k+1}|\underline{y}_{k+1}) &= \frac{p(\underline{y}_{k+1}|\underline{x}_{k+1}) p_k^x(\underline{x}_{k+1})}{\int p(\underline{y}_{k+1}|\underline{x}_k) p_k^x(\underline{x}_k) \ \mathrm{d}\underline{x}_k} \\ &= \frac{p(\underline{y}_{k+1}|\underline{x}_{k+1}) \int p(\underline{x}_{k+1}|\underline{x}_k, \underline{u}_k) p_k^x(\underline{x}_k) \ \mathrm{d}\underline{x}_k}{\int p(\underline{y}_{k+1}|\underline{x}_k) p_k^x(\underline{x}_k) \ \mathrm{d}\underline{x}_k} \ , \end{aligned}$$

where $p(\underline{y}_{k+1}|\underline{x}_k) = \int p(\underline{y}_{k+1}|\underline{x}_{k+1}) p(\underline{x}_{k+1}|\underline{x}_k, \underline{u}_k) \ \mathrm{d}\underline{x}_{k+1}$. Recursion (5) is also referred to as *value iteration*. For continuous stochastic optimal control problems, this recursion resolves to

$$\begin{aligned} V_K(p_K^x(\underline{x}_K)) &= \int \mathcal{C}_K(\underline{x}_K) p_K^x(\underline{x}_K) \ \mathrm{d}\underline{x}_K \ , \\ V_k(p_k^x(\underline{x}_k)) &= \inf_{\underline{u}_k} \int p_k^x(\underline{x}_k) \left[ \mathcal{C}_k(\underline{x}_k, \underline{u}_k) \right. \\ &\quad \left. + \int p(\underline{y}_{k+1}|\underline{x}_k) V_{k+1}(p_{k+1}^p(\underline{x}_{k+1}|\underline{y}_{k+1})) \ \mathrm{d}\underline{y}_{k+1} \right] \mathrm{d}\underline{x}_k \ . \end{aligned}$$

In the course of this paper, we will need to compute the empirical covariance of a state estimate as given in the following lemma.

**Lemma 1** *For the Dirac distribution $p_k^x(\underline{x}_k)$ as in* (4) *and the Dirac distributions $p_k^w(\underline{w}_k) = \sum_{m=1}^{N_w} \beta_k^m \delta(\underline{w}_k - \underline{w}_k^m)$ and $p_{k+1}^v(\underline{v}_{k+1}) = \sum_{n=1}^{N_v} \gamma_{k+1}^n \delta(\underline{v}_{k+1} - \underline{v}_{k+1}^n)$ of the process and the measurement noises, the empirical covariance $\mathbf{C}_{k+1}^p$ of the probability distribution $p_{k+1}^x(\underline{x}_{k+1})$ is given by*

$$\mathbf{C}_{k+1}^p = \mathbf{C}_{k+1}^{xx} - \mathbf{C}_{k+1}^{xy} (\mathbf{C}_{k+1}^{yy})^{-1} (\mathbf{C}_{k+1}^{xy})^\top \ ,$$

*where*

$$\begin{aligned} \mathbf{C}_{k+1}^{xx} &= \sum_{i=1}^{N_x} \sum_{m=1}^{N_w} \alpha_k^i \beta_k^m (\underline{x}_{k+1}^{[im]} - \widehat{\underline{x}}_{k+1}) (\underline{x}_{k+1}^{[im]} - \widehat{\underline{x}}_{k+1})^\top , \\ \mathbf{C}_{k+1}^{xy} &= \sum_{i=1}^{N_x} \sum_{m=1}^{N_w} \sum_{n=1}^{N_v} \alpha_k^i \beta_k^m \gamma_{k+1}^n (\underline{x}_{k+1}^{[im]} - \widehat{\underline{x}}_{k+1}) (\underline{y}_{k+1}^{[imn]} - \widehat{\underline{y}}_{k+1})^\top , \\ \mathbf{C}_{k+1}^{yy} &= \sum_{i=1}^{N_x} \sum_{m=1}^{N_w} \sum_{n=1}^{N_v} \alpha_k^i \beta_k^m \gamma_{k+1}^n (\underline{y}_{k+1}^{[imn]} - \widehat{\underline{y}}_{k+1}) (\underline{y}_{k+1}^{[imn]} - \widehat{\underline{y}}_{k+1})^\top , \end{aligned}$$

*with*

$$\begin{aligned} \underline{x}_{k+1}^{[im]} &= \underline{a}_k(\underline{x}_k^i, \underline{u}_k, \underline{w}_k^m) \ , \\ \widehat{\underline{x}}_{k+1}^{[im]} &= \sum\nolimits_{i=1}^{N_x} \sum\nolimits_{m=1}^{N_w} \alpha_k^i \beta_k^m \underline{x}_{k+1}^{[im]} \ , \\ \underline{y}_{k+1}^{[imn]} &= \underline{h}_{k+1}(\underline{x}_{k+1}^{[im]}, \underline{v}_{k+1}^n) \ , \\ \widehat{\underline{y}}_{k+1} &= \sum\nolimits_{i=1}^{N_x} \sum\nolimits_{m=1}^{N_w} \sum\nolimits_{n=1}^{N_v} \alpha_k^i \beta_k^m \gamma_{k+1}^n \underline{y}_{k+1}^{[imn]} \ . \end{aligned}$$

Please observe that the covariance $\mathbf{C}_{k+1}^p$ is independent of the measurement $\underline{\boldsymbol{y}}_{k+1}$.

## III. Proposed Approach

As mentioned in the introduction, we propose to use a statistical Taylor expansion of the costs-to-go in the presented local DP approach. To this end, we define this expansion as follows.

**Definition 1** *Given a Dirac distribution $\bar{p}^x(\underline{x}) = \sum_{i=1}^{N_x} \bar{\alpha}^i \delta(\underline{x} - \bar{\underline{x}}^i)$ with $N_x \in \mathbb{N}$, $\bar{\underline{x}}^i \in \mathbb{R}^{n_x}$, $\bar{\alpha}^i \in (0,1]$ for $i = 1, \ldots, N_x$, $\sum_{i=1}^{N_x} \bar{\alpha}^i = 1$, the value of a function $a(p^x(\underline{x})) = \mathrm{E}\{g(\underline{\boldsymbol{x}})\}$ of a random variable $\underline{\boldsymbol{x}}$ with probability distribution $p^x(\underline{x})$ can be approximated according to*

$$\begin{aligned} a(p^x(\underline{x})) \approx \mathrm{E}\Bigg\{ \sum\nolimits_{i=1}^{N_x} \bar{\alpha}^i \Bigg[ & g(\bar{\underline{x}}^i) + \nabla g(\bar{\underline{x}}^i)^\top (\underline{\boldsymbol{x}} - \bar{\underline{x}}^i) \\ & + \frac{1}{2} (\underline{\boldsymbol{x}} - \bar{\underline{x}}^i)^\top \nabla^2 g(\bar{\underline{x}}^i) (\underline{\boldsymbol{x}} - \bar{\underline{x}}^i) \Bigg] \Bigg\} \ , \end{aligned} \tag{6}$$

*if the distributions $p^x(\underline{x})$ and $\bar{p}^x(\underline{x})$ are independent and close*[2].

[2]To determine the closeness of the two probability distributions $p^x(\underline{x})$ and $\bar{p}^x(\underline{x})$, we can either use a distance measure for probability distributions [24] or the absolute difference between $a(\bar{p}^x(\underline{x}))$ and the approximation $a(p^x(\underline{x}))$.

Equation (6) can be derived using an approach similar to the approximate computation of the expected value of a random variable. First, we use

$$a(p^x(\underline{x})) = \mathrm{E}\{g(\underline{\boldsymbol{x}})\} = \mathrm{E}\Big\{g(\underline{\boldsymbol{y}} + \underline{\boldsymbol{x}} - \underline{\boldsymbol{y}})\Big\} \ ,$$

where $\underline{\boldsymbol{y}}$ is an arbitrary random variable. Next, we expand the expected value and perform a second-order Taylor expansion of $g(\cdot)$ at $\underline{\boldsymbol{y}}$, which gives

$$\begin{aligned} &a(p^x(\underline{x})) \\ &\approx \mathrm{E}\Big\{g(\underline{\boldsymbol{y}}) + \nabla g(\underline{\boldsymbol{y}})^\top(\underline{\boldsymbol{x}} - \underline{\boldsymbol{y}}) + \frac{1}{2}(\underline{\boldsymbol{x}} - \underline{\boldsymbol{y}})^\top \nabla^2 g(\underline{\boldsymbol{y}})(\underline{\boldsymbol{x}} - \underline{\boldsymbol{y}})\Big\} \\ &= \int\int \Big[g(\underline{y}) + \nabla g(\underline{y})^\top(\underline{x} - \underline{y}) + \frac{1}{2}(\underline{x} - \underline{y})^\top \nabla^2 g(\underline{y})(\underline{x} - \underline{y})\Big] \\ &\qquad \times p(\underline{x}, \underline{y}) \ \mathrm{d}\underline{x} \ \mathrm{d}\underline{y} \ . \end{aligned}$$

Here, $p(\underline{x}, \underline{y})$ is a joint distribution of $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$. Then, assuming that $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ are independent and that $\bar{p}^x(\underline{y})$ is the probability distribution of $\underline{\boldsymbol{y}}$, we obtain (6).

Using Definition 1, we can perform statistical second-order approximation of the costs-to-go along a reference trajectory formed by the Dirac distributions $\bar{p}^x_{0:K}(\underline{x}_{0:K})$ and the control inputs $\bar{\underline{u}}_{0:K-1}$. To obtain a reference trajectory given an initial policy $\underline{\pi}_{0:K-1}(p^x_{0:K-1}(\underline{x}_{0:K-1}))$, we proceed as follows. Of course, we can use the nominal trajectory as

**Algorithm 1** Generation of a reference trajectory.

*Step 1:* Set $k = 0$ and initialize $\bar{p}^x_0(\underline{x}_0) = p^x_0(\underline{x}_0)$.
*Step 2:* Compute $\bar{\underline{u}}_k = \underline{\pi}_k(\bar{p}^x_k(\underline{x}_k))$.
*Step 3:* Use a sample from $\bar{p}^x_k(\underline{x}_k)$, $\bar{\underline{u}}_k$ from *Step 2*, and a process noise sample from $p^w_k(\underline{w}_k)$ in order to obtain a simulated system state $\underline{x}_{k+1}$.
*Step 4:* With $\underline{x}_{k+1}$ from *Step 3* and a measurement noise sample from $p^v_{k+1}(\underline{v}_{k+1})$, generate a simulated observation $\underline{y}_{k+1}$.
*Step 5:* With the measurement $\underline{y}_{k+1}$, compute the state estimate $\bar{p}^x_{k+1}(\underline{x}_{k+1})$.
*Step 6:* If $k = K$, stop the algorithm. Otherwise, set $k = k + 1$ and return to *Step 2*.

the reference trajectory. To generate the nominal trajectory, we use the means of $p^x_k(\underline{x}_k)$, $p^w_k(\underline{w}_k)$, and $p^v_{k+1}(\underline{v}_{k+1})$ in the above algorithm.

Next, we address statistical Taylor expansion of the value functions using the method from Definition 1. The approximation of the costs-to-go at time step $K$ can be obtained according to the following theorem.

**Theorem 1** *Using the approximation method from Definition 1, a second-order approximation of the costs-to-go $V_K(p^x_K(\underline{x}_K))$ at the reference state estimate $\bar{p}^x_K(\underline{x}_K)$ computes to*

$$V_K(p^x_K(\underline{x}_K)) = \mathrm{E}\Big\{s_K + \mathbf{v}_K^\top \underline{\boldsymbol{x}}_K + \frac{1}{2}\underline{\boldsymbol{x}}_K^\top \mathbf{V}_K \underline{\boldsymbol{x}}_K + \frac{1}{2}\mathrm{tr}[\mathbf{S}_K \mathbf{C}_K]\Big\} ,$$

*where*

$$\begin{aligned} s_K &= \sum_{i=1}^{N_x} \bar{\alpha}^i_K \Big[\mathcal{C}_K(\bar{\underline{x}}^i_K) - \nabla\mathcal{C}_K(\bar{\underline{x}}^i_K)^\top \bar{\underline{x}}^i_K + \frac{1}{2}(\bar{\underline{x}}^i_K)^\top \nabla^2 \mathcal{C}_K(\bar{\underline{x}}^i_K)\bar{\underline{x}}^i_K\Big] , \\ \mathbf{v}_K &= \sum_{i=1}^{N_x} \bar{\alpha}^i_K \left[\nabla\mathcal{C}_K(\bar{\underline{x}}^i_K) - \nabla^2\mathcal{C}_K(\bar{\underline{x}}^i_K)\bar{\underline{x}}^i_K\right] , \\ \mathbf{V}_K &= \sum_{i=1}^{N_x} \bar{\alpha}^i_K \nabla^2 \mathcal{C}_K(\bar{\underline{x}}^i_K) \ , \qquad \mathbf{S}_K = \mathbf{0} \ . \end{aligned} \tag{7}$$

*Proof:* We have $V_K(p^x_K(\underline{x}_K)) = \mathrm{E}\{\mathcal{C}_K(\underline{\boldsymbol{x}}_K)\}$ according to (5) and apply the approximation from Definition 1 with $g(\underline{\boldsymbol{x}}_K) = \mathcal{C}_K(\underline{\boldsymbol{x}}_K)$. By separating the constant, linear, and quadratic terms, we can obtain the above result. ■

In the next theorem, we present an approximation of the costs-to-go along a reference trajectory at time step $k$.

**Theorem 2** *Using the result from Theorem 1 and Definition 1, a second-order statistical approximation of the costs-to-go $V_k(p^x_k(\underline{x}_k))$ at the reference state estimate $\bar{p}^x_k(\underline{x}_k)$ and a reference control input $\bar{\underline{u}}_k$ can be computed according to*

$$V_k(p^x_k(\underline{x}_k)) = \mathrm{E}\Big\{s_k + \mathbf{v}_k^\top \underline{\boldsymbol{x}}_k + \frac{1}{2}\underline{\boldsymbol{x}}_k^\top \mathbf{V}_k \underline{\boldsymbol{x}}_k + \frac{1}{2}\mathrm{tr}[\mathbf{S}_k \mathbf{C}_k]\Big\} ,$$

*with $\mathbf{C}_k$ being the empirical covariance of $p^x_k(\underline{x}_k)$ (see Lemma 1) and*

$$\begin{aligned} s_k &= \left(\frac{\epsilon_k^2}{2} - \epsilon_k\right) \underline{r}_k^\top \mathbf{R}_k^{-1} \underline{r}_k - \frac{1}{2}\widetilde{\underline{p}}_k^\top \mathbf{R}_k^{-1} \widetilde{\underline{p}}_k + \underline{r}_k^\top \mathbf{R}_k^{-1} \widetilde{\underline{p}}_k \\ &\quad + \varphi_k(\bar{\underline{x}}_k^{1:N_x}, \bar{\underline{u}}_k) + \sum_{i=1}^{N_x} \left[\frac{1}{2}(\bar{\underline{x}}^i_k)^\top \widetilde{\underline{q}}^i_k - (\underline{p}^i_k)^\top \bar{\underline{x}}^i_k\right] \ , \\ \mathbf{v}_k &= \widetilde{\mathbf{P}}_k^\top \mathbf{R}_k^{-1} \widetilde{\underline{p}}_k - \widetilde{\mathbf{P}}_k^\top \mathbf{R}_k^{-1} \underline{r}_k + \sum_{i=1}^{N_x} \left[\underline{p}^i_k - \widetilde{\underline{q}}^i_k\right] \ , \\ \mathbf{V}_k &= -\widetilde{\mathbf{P}}_k^\top \mathbf{R}_k^{-1} \widetilde{\mathbf{P}}_k + \widetilde{\mathbf{Q}}_k \ , \quad \mathbf{S}_k = \widetilde{\mathbf{P}}_k^\top \mathbf{R}_k^{-1} \widetilde{\mathbf{P}}_k \ , \end{aligned} \tag{8}$$

*where $\epsilon_k \in (0, 1]$ is a parameter and*

$$\begin{aligned} \varphi_k(\bar{\underline{x}}_k^{1:N_x}, \bar{\underline{u}}_k) &= \sum_{i=1}^{N_x} \sum_{m=1}^{N_w} \bar{\alpha}^i_k \bar{\beta}^m_k \Big[\mathcal{C}_k(\bar{\underline{x}}^i_k, \bar{\underline{u}}_k) + s_{k+1} \\ &\quad + \mathbf{v}_{k+1}^\top \underline{a}_k(\bar{\underline{x}}^i_k, \bar{\underline{u}}_k, \bar{\underline{w}}^m_k) + \frac{1}{2}\mathrm{tr}\left[\mathbf{S}_{k+1}\mathbf{C}^p_{k+1}\right] \\ &\quad + \frac{1}{2}\underline{a}_k(\bar{\underline{x}}^i_k, \bar{\underline{u}}_k, \bar{\underline{w}}^m_k)^\top \mathbf{V}_{k+1} \underline{a}_k(\bar{\underline{x}}^i_k, \bar{\underline{u}}_k, \bar{\underline{w}}^m_k)\Big] , \\ p_k &= \begin{bmatrix} p^1_k \\ \vdots \\ p^{N_x}_k \end{bmatrix} \ , \ \begin{bmatrix} \underline{p}^i_k \\ \underline{r}_k \end{bmatrix} = \nabla\varphi_k(\bar{\underline{x}}_k^{1:N_x}, \bar{\underline{u}}_k) \ , \\ \begin{bmatrix} \mathbf{Q}^{11}_k & \dots & \mathbf{Q}^{1N_x}_k & (\mathbf{P}^1_k)^\top \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{Q}^{N_x 1}_k & \dots & \mathbf{Q}^{N_x N_x}_k & (\mathbf{P}^{N_x}_k)^\top \\ \mathbf{P}^1_k & \dots & \mathbf{P}^{N_x}_k & \mathbf{R}_k \end{bmatrix} &= \nabla^2\varphi_k(\bar{\underline{x}}_k^{1:N_x}, \bar{\underline{u}}_k) \ , \\ \widetilde{\underline{p}}_k = \sum\nolimits_{i=1}^{N_x} \mathbf{P}^i_k \bar{\underline{x}}^i_k \ , \ \widetilde{\mathbf{P}}_k &= \sum\nolimits_{i=1}^{N_x} \mathbf{P}^i_k \ , \\ \widetilde{\underline{q}}^i_k = \sum\nolimits_{j=1}^{N_x} \mathbf{Q}^{ij}_k \bar{\underline{x}}^j_k \ , \ \widetilde{\mathbf{Q}}_k &= \sum\nolimits_{i=1}^{N_x} \sum\nolimits_{j=1}^{N_x} \mathbf{Q}^{ij}_k \ , \end{aligned} \tag{9}$$

*where we write $\mathbf{C}^p_{k+1}$ to indicate that the covariance $\mathbf{C}_{k+1}$ is computed using $p^x_k(\underline{x}_k)$ and not $p^x_{k+1}(\underline{x}_{k+1})$.*

*Proof:* We will prove Theorem 2 by induction. The costs-to-go at time step $k$, when using the approximation from Theorem 2 for the value function $V_{k+1}(\underline{x}_{k+1})$, are given by

$$\begin{aligned}V_k(p_k^x(\underline{x}_k)) &= \inf_{\underline{u}_k} \mathrm{E}\big\{\mathcal{C}_k(\underline{\boldsymbol{x}}_k,\underline{u}_k) + s_{k+1} + \mathbf{v}_{k+1}^\top \underline{\boldsymbol{x}}_{k+1} \\ &+\frac{1}{2}\underline{\boldsymbol{x}}_{k+1}^\top \mathbf{V}_{k+1}\underline{\boldsymbol{x}}_{k+1} + \frac{1}{2}\mathrm{tr}[\mathbf{S}_{k+1}\mathbf{C}_{k+1}]\Big\} \\ &= \inf_{\underline{u}_k} \mathrm{E}\big\{\mathcal{C}_k(\underline{\boldsymbol{x}}_k,\underline{u}_k) + s_{k+1} + \mathbf{v}_{k+1}^\top \underline{a}_k(\underline{\boldsymbol{x}}_k,\underline{u}_k,\underline{\boldsymbol{w}}_k) \\ &+\frac{1}{2}\underline{a}_k(\underline{\boldsymbol{x}}_k,\underline{u}_k,\underline{\boldsymbol{w}}_k)^\top \mathbf{V}_{k+1}\underline{a}_k(\underline{\boldsymbol{x}}_k,\underline{u}_k,\underline{\boldsymbol{w}}_k) + \frac{1}{2}\mathrm{tr}\left[\mathbf{S}_{k+1}\mathbf{C}_{k+1}^p\right]\Big\}.\end{aligned}$$

Next, applying the expansion from Definition 1 at $\bar{p}_k^x(\underline{x}_k)$ yields

$$\begin{aligned}V_k(p_k^x(\underline{x}_k)) \approx \inf_{\underline{u}_k} \mathrm{E}\Bigg\{&\varphi_k(\bar{\underline{x}}_k^{1:N_x},\bar{\underline{u}}_k) + \begin{bmatrix}\underline{p}_k^1\\ \vdots \\ \underline{p}_k^{N_x}\\ \underline{r}_k\end{bmatrix}^\top \begin{bmatrix}\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^1\\ \vdots \\ \underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^{N_x}\\ \underline{u}_k - \bar{\underline{u}}_k\end{bmatrix} \\ &+\frac{1}{2}\begin{bmatrix}\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^1\\ \vdots \\ \underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^{N_x}\\ \underline{u}_k - \bar{\underline{u}}_k\end{bmatrix}^\top \begin{bmatrix}\mathbf{Q}_k^{11} & \dots & \mathbf{Q}_k^{1N_x} & (\mathbf{P}_k^1)^\top\\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{Q}_k^{N_x 1} & \dots & \mathbf{Q}_k^{N_x N_x} & (\mathbf{P}_k^{N_x})^\top \\ \mathbf{P}_k^1 & \dots & \mathbf{P}_k^{N_x} & \mathbf{R}_k\end{bmatrix} \\ &\times \begin{bmatrix}\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^1\\ \vdots \\ \underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^{N_x}\\ \underline{u}_k - \bar{\underline{u}}_k\end{bmatrix}\Bigg\}. \end{aligned} \tag{10}$$

Please note that we needed to apply Definition 1 twice because the covariance-related term contains inner integrals due to evaluation of the inner expected values in $\mathbf{C}_{k+1} = \mathrm{E}\big\{(\underline{\boldsymbol{x}}_{k+1} - \mathrm{E}\big\{\underline{\boldsymbol{x}}_{k+1}\big\})(\underline{\boldsymbol{x}}_{k+1} - \mathrm{E}\big\{\underline{\boldsymbol{x}}_{k+1}\big\})^\top\big\}$.

Evaluation of the necessary optimality condition with respect to $\underline{u}_k$ for the above equation gives us

$$\underline{u}_k = -\mathbf{R}_k^{-1}\widetilde{\mathbf{P}}_k \mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \mathbf{R}_k^{-1}\widetilde{\underline{p}}_k - \mathbf{R}_k^{-1}\underline{r}_k + \bar{\underline{u}}_k\ .$$

At this point, we introduce the parameter $\epsilon_k \in (0,1]$, as proposed in [18] and obtain

$$\underline{u}_k = -\mathbf{R}_k^{-1}\widetilde{\mathbf{P}}_k \mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \mathbf{R}_k^{-1}\widetilde{\underline{p}}_k - \epsilon_k\mathbf{R}_k^{-1}\underline{r}_k + \bar{\underline{u}}_k\ . \tag{11}$$

We will use the parameter $\epsilon_k$ in order to ensure convergence of the algorithm for the computation of the controller parameters $\mathbf{L}_k$ and $\underline{d}_k$ that we present later. Now, using (11) in (10) yields

$$\begin{aligned}V_k(p_k^x(\underline{x}_k)) \approx \mathrm{E}\Bigg\{&\varphi_k(\bar{\underline{x}}_k^{1:N_x},\bar{\underline{u}}_k) + \sum\nolimits_{i=1}^{N_x}\Bigg[\left(\underline{p}_k^i\right)^\top(\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^i) \\ &+\sum\nolimits_{j=1}^{N_x}\frac{1}{2}(\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^i)^\top\mathbf{Q}_k^{ij}\left(\underline{\boldsymbol{x}}_k - \bar{\underline{x}}_k^{ij}\right)\Bigg] + \underline{r}_k^\top(\underline{u}_k - \bar{\underline{u}}_k) \\ &+\left(\widetilde{\mathbf{P}}_k\mathrm{E}\{\underline{\boldsymbol{x}}_k\} - \widetilde{\underline{p}}_k\right)^\top(\underline{u}_k - \bar{\underline{u}}_k) \\ &+\frac{1}{2}\left(-\widetilde{\mathbf{P}}_k\mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \widetilde{\underline{p}}_k - \epsilon_k\underline{r}_k\right)^\top(\underline{u}_k - \bar{\underline{u}}_k)\Bigg\}\ .\end{aligned}$$

After a manipulation, we obtain the result from Theorem 2. To this end, we use that for a deterministic matrix $\mathbf{A}$ it holds $\mathrm{E}\{\underline{\boldsymbol{x}}\mathbf{A}\underline{\boldsymbol{x}}\} = \mathrm{tr}\big[\mathbf{A}\mathrm{E}\big\{(\underline{\boldsymbol{x}} - \mathrm{E}\{\underline{\boldsymbol{x}}\})(\underline{\boldsymbol{x}} - \mathrm{E}\{\underline{\boldsymbol{x}}\})^\top\big\}\big] + \mathrm{E}\big\{\underline{\boldsymbol{x}}_k^\top\big\}\mathbf{A}\mathrm{E}\{\underline{\boldsymbol{x}}\}$, where $\mathrm{E}\big\{(\underline{\boldsymbol{x}} - \mathrm{E}\{\underline{\boldsymbol{x}}\})(\underline{\boldsymbol{x}} - \mathrm{E}\{\underline{\boldsymbol{x}}\})^\top\big\}$ can be identified as the covariance of $\underline{\boldsymbol{x}}$. The approximation of $V_k(p_k^x(\underline{x}_k))$ is independent of $\underline{\boldsymbol{y}}_{k+1}$ if computed according to Lemma 1. ■

Please note that in Theorem 2, the matrix $\mathbf{R}_k$ is positive definite and thus invertible due to the assumptions in (3).

With Theorems 1 and 2, we can now formulate the procedure for the computation of the controller parameters $\mathbf{L}_{0:K-1}$ and $\underline{d}_{0:K-1}$ as given in Algorithm 2. The backtracking line search in Step 4 ensures the convergence of the costs [18]. The search is performed by setting $\epsilon_k = 1$. If the costs-to-go $V_k(\bar{p}_k^x(\underline{x}_k))$ for $\underline{u}_k$ computed according to (11) are larger than the costs-to-go for $\underline{u}_k = \mathbf{L}_k^{[\eta]}\mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \underline{d}_k^{[\eta]}$, we set $\epsilon_k = \epsilon_k/2$ and repeat. If $\epsilon_k$ at a time step $k$ becomes smaller than a predefined positive threshold, the current control law for this time step is locally optimal.

**Algorithm 2** Computation of the controller parameters.

*Step 1:* Select an initial control law $\mathbf{L}_{0:K-1}^{[0]}$, $\underline{d}_{0:K-1}^{[0]}$ and set $\eta = 0$.

*Step 2:* Use the current control law $\mathbf{L}_{0:K-1}^{[\eta]}$, $\underline{d}_{0:K-1}^{[\eta]}$ and the initial state estimate $p_0^x(\underline{x})$ in Algorithm 1 in order to generate a trajectory of closed-loop reference state estimates $\bar{p}_{0:K}^x(\underline{x}_{0:K})$ and reference control inputs $\bar{\underline{u}}_{0:K-1}$ by interleaved filtering and policy evaluation.

*Step 3:* Initialize $s_K, \mathbf{v}_K, \mathbf{V}_K$, and $\mathbf{S}_K$ according to (7), and set $k = K$.

*Step 4:* Set $k = k-1$. Then:

1) Compute $\mathbf{R}_k$, $\widetilde{\mathbf{P}}_k$, $\underline{r}_k$, and $\widetilde{\underline{p}}_k$ using (9).
2) Determine $\epsilon_k$ using backtracking line search initialized with $\epsilon_k = 1$ such that the costs-to-go for $\underline{u}_k$ computed using (11) become smaller than the costs-to-go for $\underline{u}_k = \mathbf{L}_k^{[\eta]}\mathrm{E}\{\underline{\boldsymbol{x}}_k\} + \underline{d}_k^{[\eta]}$.
3) Set $\mathbf{L}_k^{[\eta+1]} = -\mathbf{R}_k^{-1}\widetilde{\mathbf{P}}_k$, $\underline{d}_k^{[\eta+1]} = \mathbf{R}_k^{-1}(\widetilde{\underline{p}}_k - \epsilon_k\underline{r}_k) + \bar{\underline{u}}_k$, and compute $s_k$, $\mathbf{v}_k$, $\mathbf{V}_k$, and $\mathbf{S}_k$ using (8).
4) If $k = 0$, proceed to *Step 5* and return to *Step 4* otherwise.

*Step 5:* If the costs converged, i.e., if $|\mathcal{J}^{[\eta+1]} - \mathcal{J}^{[\eta]}| \leq \alpha$, where $\alpha$ is a small positive constant, stop the algorithm. Otherwise, set $\eta = \eta + 1$ and return to *Step 2*.

## IV. Numerical Example

We demonstrate the proposed control approach in a numerical toy example. In our scenario, a robot with simple linear dynamics $\underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k + \underline{u}_k + \underline{\boldsymbol{w}}_k$ with $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, 0.5^2\mathbf{I})$ has to reach $\underline{x}_T = \begin{bmatrix}-16 & 16\end{bmatrix}^\top$ starting with the initial Gaussian state estimate

$$p_0^x(\underline{x}_0) = \mathcal{N}\left(\begin{bmatrix}-4\\ 4\end{bmatrix}, \begin{bmatrix}3 & -0.5\\ -0.5 & 3\end{bmatrix}\right).$$

The measurements fed back to the controller are obtained according to

$$\underline{\boldsymbol{y}}_k = \begin{bmatrix}\|\underline{\boldsymbol{x}}_k - \underline{z}_1\|^2\\ \|\underline{\boldsymbol{x}}_k - \underline{z}_2\|^2\end{bmatrix} + \begin{bmatrix}\|\underline{\boldsymbol{x}}_k - \underline{z}_1\|^2\underline{\boldsymbol{v}}_{k,1}\\ \|\underline{\boldsymbol{x}}_k - \underline{z}_2\|^2\underline{\boldsymbol{v}}_{k,2}\end{bmatrix},$$
$$\begin{bmatrix}\underline{\boldsymbol{v}}_{k,1}\\ \underline{\boldsymbol{v}}_{k,2}\end{bmatrix} \sim \mathcal{N}(\underline{0}, 0.05^2\mathbf{I})$$

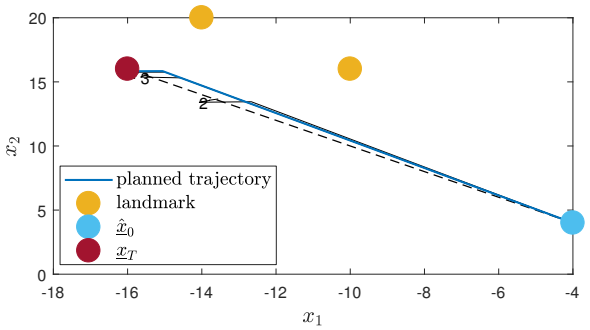



Fig. 2: Planning result after 5 iterations.

i.e., the robot receives distance measurements to $\underline{z}_1$ and $\underline{z}_2$ that are subject to state-dependent measurement noise. In the considered scenario, we set the landmarks $\underline{z}_1 = \begin{bmatrix}-10 & 16\end{bmatrix}^\top$ and $\underline{z}_2 = \begin{bmatrix}-14 & 20\end{bmatrix}^\top$. The cost function is assumed to be

$$\mathcal{J} = \mathrm{E}\left\{20\left\|\underline{\boldsymbol{x}}_K - \underline{x}_T\right\|^2 + \sum_{k=0}^{K-1}\left(20\left\|\underline{\boldsymbol{x}}_k - \underline{x}_T\right\|^2 + \left\|\underline{u}_k\right\|^2\right)\right\},$$

where $K$ denotes the length of the planning horizon that is set $K = 10$ in the simulation. As a state estimator, we use the UKF and compute the gradient and the Hessian of $\varphi_k$ from (9) using adaptive numerical differentiation.

If we applied the LQG in the considered scenario, it would plan a straight line from the mean $\hat{\underline{x}}_0$ of $p_0^x(\underline{x}_0)$ to $\underline{x}_T$, because it assumes separation between control and estimation. In contrast, our control approach considers that the state estimation quality near to one of the landmarks gets better. Therefore, it balances the costs induced by poor estimation quality and the costs induced by the control. This expectation can be seen in Fig. 2 that depicts the state trajectory planned by the proposed algorithm after 10 iterations. It can be seen that the controller deforms the path from $\hat{\underline{x}}_0$ to $\underline{x}_T$ towards the landmarks in order to obtain a more precise state estimate.

## V. Conclusion

In this paper, we presented a trajectory optimization algorithm for closed-loop control of stochastic nonlinear systems. The proposed method relies on statistical second-order approximation of the costs-to-go along a reference trajectory of closed-loop state estimates that are maintained in terms of Dirac distributions. In contrast to state-of-the-art approaches that rely on EKF-based approximation of the costs-to-go, we expect our method to be more robust. Furthermore, by using a sample-based representation of the state estimates, we are able to deal with non-Gaussian system states.

Our future work consists in evaluation of the proposed algorithm with more sophisticated filters such as the particle filter, and its comparison with related state-of-the-art approaches from [17] and [18]. Furthermore, we plan to drop the assumption of an affine controller by using nonlinear policies and Q-learning. Finally, an extension of the proposed approach to problems with chance constraints is also planned.

## References


[1] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, Belmont, Massachusetts, 2000, vol. 1.

[2] R. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory*. New York, USA: Academic Press, 1965.

[3] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley and Sons, Inc., 1994.

[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, Dec 1971.

[5] Y. Bar-Shalom and E. Tse, "Dual Effect, Certainty Equivalence, and Separation in Stochastic Control," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 494–500, Oct 1974.

[6] 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.

[7] S. Thrun, "Monte Carlo POMDPs," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K.-R. Müller, Eds. MIT Press, 2000, pp. 1064–1070.

[8] R. A. L. S. Kullback, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 79–86, 1951.

[9] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[10] R. Smallwood 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.

[11] 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.

[12] 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.

[13] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Monte Carlo Value Iteration for Continuous-State POMDPs, pp. 175–191.

[14] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte, "Parametric POMDPs for Planning in Continuous State Spaces," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 887 – 897, 2006.

[15] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," Jun. 2005.

[16] D. Mayne, "A Second-order Gradient Method for Determining Optimal Trajectories of Non-linear Discrete-time Systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

[17] 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.

[18] J. van den Berg, S. Patil, and R. Alterovitz, "Motion Planning under Uncertainty using Iterative Local Optimization in Belief Space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[19] A. Bry and N. Roy, "Rapidly-exploring Random Belief Trees for Motion Planning Under Uncertainty," 2011.

[20] T. Lefebvre, H. Bruyninckx, and J. de Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics. Springer, 2005.

[21] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[22] J. Duník, O. Straka, and M. Simandl, "The Development of a Randomised Unscented Kalman Filter," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8–13, 2011.

[23] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Qnline Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

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