Open-Loop Feedback Control of Nonlinear Stochastic Systems Based on Deterministic Dirac Mixture Densities

Achim Hekler, Christof Chlebek, and Uwe D. Hanebeck

***Abstract*— The main problem of stochastic nonlinear model predictive control (SNMPC) is that the equations for state prediction and calculation of the expected reward are in general not solvable in closed form. A popular approach is to approximate the occurring continuous probability density functions by a discrete density representation, which allows an analytical solution of the SNMPC equations. In this paper, we propose to draw the samples not randomly as in Monte Carlo based methods, but systematically by minimizing a distance measure. In doing so, fewer components are generally required to represent the underlying probability density while achieving the same approximation quality. Especially if the evaluation of the expected reward is computationally expensive, this property affects the complexity of computation significantly. By means of a path planning problem, we have substantiated this statement with several simulation runs.**

I. INTRODUCTION

In model predictive control, the controller uses a model of the system in order to incorporate not only the current system state, but also the future behavior of the system into the decision about the current optimal control input. This fact often results in a higher quality of control and therefore, model predictive controllers are already applied in many areas [1].

The more closely the model reflects the real-world process, the more reliable is the prediction of future system states. Therefore, it is particularly desirable for strongly nonlinear systems that the applied controller can deal with nonlinear system models.

In addition to a precise model of the system under control, occurring uncertainties should also be incorporated in the control decision. These uncertainties include exogenous and endogenous disturbances acting upon the system. Moreover, sources of uncertainties might be noisy sensor measurements, which lead to an uncertain state estimate.

Since a deterministic approach does not allow a complete modeling of all occurring phenomena for reasons of complexity, it is popular to model these uncertainties probabilistically, leading to *stochastic nonlinear model predictive control* (SNMPC).

In this paper, we present an SNMPC method for systems with continuous-valued system states and a finite set of discrete control inputs. This system class represents many technical systems. For example, the state space of the walking robots introduced in [2] comprises the continuous position. In addition, this kind of robot is only able to handle a finite set of discrete control inputs, namely commands such as turn left/right or move forward.

The main problem of SNMPC is that the equations for state prediction and calculation of the expected reward are not generally solvable in closed form. One possibility to solve this problem is to approximate the occurring densities by an adequate density representation, for which the SNMPC equations can be solved analytically. For example, a popular approach is to represent the densities and the reward function with Gaussian Mixtures [2], [3], [4].

The advantages of discrete representations of an underlying continuous probability density are that both, the probabilistic prediction of the current state estimate as well as the evaluation of the reward function rating the extrapolated states can be easily realized.

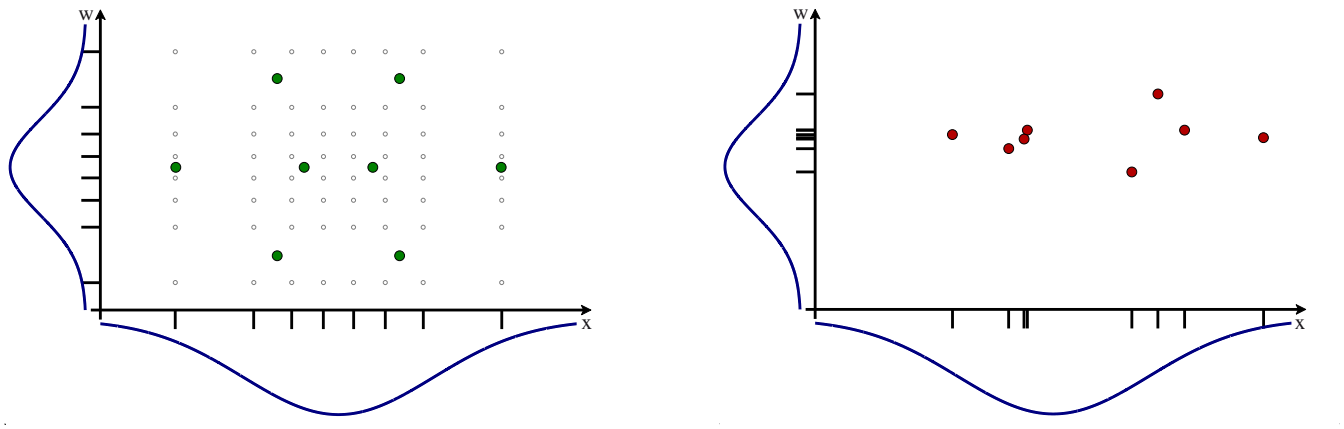
A. Contributions

In contrast to the approach proposed in this paper, existing sample-based SNMPC approaches determine the particles in a purely random manner [5], [6], [7]. The key idea of this paper is to approximate the occuring densities by a discrete representation of an underlying continuous density, a *deterministic Dirac mixture density*. In contrast to Monte Carlo methods, where the particles are drawn randomly, the components of the Dirac mixture density are determined systematically by optimizing a distance measure. Fig. 1 illustrates the difference between the systematic and the Monte Carlo approach. In doing so, the underlying density can be generally represented by fewer components while achieving the same approximation quality. This property is advantageous in the SNMPC context, since the complexity of the control decision depends directly on the number of components representing the density. Thus, a lower complexity in run-time and storage can be attained, particularly if the evaluation of the reward function is computationally expensive.

B. Outline of this Paper

The remainder of this paper is structured as follows: in the next Section, the considered problem class of SNMPC is precisely specified. A brief introduction to deterministic Dirac mixture densities is given in Sec. III. The integration of this density representation in the SNMPC framework is the topic of Sec. IV. Finally, we illustrate the benefits of this density representation for SNMPC by various simulation

A. Hekler, C. Chlebek, and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, achim.hekler@kit.edu, christof.chlebek@kit.edu, uwe.hanebeck@ieee.org.



(a) Deterministic approximation

(b) Monte Carlo sampling

Fig. 1. This figure illustrates, how a joint probability density can be approximated in a systematic way (a) or by Monte Carlo sampling (b). The initial densities are illustrated by blue lines and their discrete approximations by short black lines on the axes. Subfigure (a) shows the approximation with a deterministic Dirac mixture density by green dots. The original Cartesian product of all Dirac components is displayed by small grey dots. For comparison, subfigure (b) shows an exemplary Monte Carlo sampling of the joint probability function depicted by red dots.

runs of a path planning problem in Sec. V. The paper concludes with a summary.

C. General Notation

Throughout this paper, random variables $\boldsymbol{x}$ are written in bold face letters, whereas deterministic quantities x are in normal letters. In order to identify vector-valued quantities $\underline{\boldsymbol{x}}$ or $\underline{x}$, we underline the corresponding identifier. The notation $\underline{\boldsymbol{x}}_k \sim f_k^x(\underline{x}_k)$ means that the random variable $\underline{\boldsymbol{x}}_k$ is characterized by its probability distribution $f_k^x(\underline{x}_k)$. Finally, a density represented by a deterministic Dirac mixture, is denoted by $\tilde{f}_k^x(\underline{x}_k)$ and a sequence $[x_a, x_{a+1}, \ldots, x_b]$ by $x_{a:b}$.

II. PROBLEM FORMULATION

In this paper, we consider a discrete-time stochastic nonlinear time-variant dynamic system

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k, \underline{\boldsymbol{w}}_k) \ , \tag{1}$$

where the system state $\underline{\boldsymbol{x}}_k$ is assumed to be not directly accessible, but can be estimated by means of noisy measurements. Hence, it is modeled by a random variable characterized by a probability density function $f_k^x(\underline{x}_k)$. The identifier $\underline{u}_k \in \mathcal{U}_k$ denotes the deterministic control input, where $\mathcal{U}_k$ is a finite set of discrete control inputs. Furthermore, $a_k(\cdot)$ is the nonlinear system function describing the dynamic behavior of the system. Finally, the random variable $\underline{\boldsymbol{w}}_k \sim f_k^w(\underline{w}_k)$ subsumes endogenous and exogenous disturbances acting upon the system.

The stochastic nonlinear system (1) is controlled by an open-loop feedback controller (OLFC) with a finite horizon, which determines the current optimal control input $\underline{u}_k^*$ as if no further measurements will be received in future time steps [8].

More precisely, $\underline{u}_k^*$ is determined as follows:

1) Based on available measurements and previously applied control inputs, the current system state $\underline{\boldsymbol{x}}_k$ is estimated.
2) Then, the OLFC determines a control sequence $\underline{u}_{k:k+H-1}^*$ that solves the open-loop problem

$$\max_{\underline{u}_{k:k+H-1}} E\left\{ g_{k+H}(\underline{\boldsymbol{x}}_{k+H}) + \sum_{i=k}^{k+H-1} g_i(\underline{\boldsymbol{x}}_i, \underline{u}_i) \right\} \tag{2}$$

subject to

$$\begin{aligned} &\underline{\boldsymbol{x}}_{i+1} = \underline{a}_i\left(\underline{\boldsymbol{x}}_i, \underline{u}_i, \underline{\boldsymbol{w}}_i\right) \text{ with} \\ &i = k, k+1, \ldots, k+H-1 \text{ and} \\ &\underline{u}_i \in \mathcal{U}_i \ . \end{aligned}$$

3) Finally, the control input $\underline{u}_k^*$ is applied to the system.

In (2), the function $g_i(\cdot)$ denotes the application-specific reward function that assigns each sequence $\underline{\boldsymbol{x}}_{k:k+H}$ of predicted system states and the corresponding sequence $\underline{u}_{k:k+H-1}$ of control inputs a real number. This function models the desired behavior of the system by assigning a high reward to preferred system states and control inputs, while undesired states and inputs obtain a low reward. In this work, we make no assumptions on the reward function and hence, it can be an *arbitrary* function.

The extrapolated future system states $\underline{\boldsymbol{x}}_i$ of (2) are calculated by means of the Chapman-Kolmogorov equation [9]

$$f_{i+1}^x(\underline{x}_{i+1}) = \int f_{\underline{u}_i}^T(\underline{x}_{i+1}|\underline{x}_i) f_i^x(\underline{x}_i)\, d\underline{x}_i \ , \tag{3}$$

where $f_{\underline{u}_i}^T(\cdot|\cdot)$ denotes the transition density and is defined by

$$\begin{aligned} &f_{\underline{u}_i}^T(\underline{x}_{i+1}|\underline{x}_i) = \\ &\int \delta(\underline{x}_{i+1} - \underline{a}_k(\underline{x}_i, \underline{u}_i, \underline{w}_i)) \cdot f_i^w(\underline{w}_i)\, d\underline{w}_i \ . \end{aligned}$$

The transition density can be interpreted as a probabilistic description of the dynamic system behavior.

There are two main challenges of stochastic OLFC:

1) The calculation of the expected value in (2) is generally not solvable in closed form for arbitrary state densities and reward functions.
2) In addition, the extrapolated system state $\underline{\boldsymbol{x}}_{k:k+H}$ cannot be determined analytically in general.

Thus, it is mandatory to either approximate the system model or the probability density functions.

III. DETERMINISTIC DIRAC MIXTURE APPROXIMATION

As mentioned before, we propose to represent the densities by a special kind of density representation, the *Dirac mixture densities*.

An arbitrary N-dimensional probability density function $f(\underline{x}) : \mathbb{R}^N \rightarrow \mathbb{R}_+$ characterizing the random vector $\underline{\boldsymbol{x}} \in \mathbb{R}^N$ can be approximated by a Dirac mixture density given by

$$\tilde{f}(\underline{x}) = \sum_{j=1}^{L} \omega^{(j)} \cdot \delta(\underline{x} - \underline{x}^{(j)}) \ , \tag{4}$$

where $\underline{x}^{(j)}$, for $j = 1, ..., L$, denotes the position of the jth Dirac component and $\omega^{(j)}$, with

$$\sum_{j=1}^{L} \omega^{(j)} = 1 \ ,$$

describes the corresponding weight.

In order to measure the quality of a density approximation, the distance between two probability densities must be defined. Common distance measures based on probability densities cannot be applied directy in the case of discrete density representations. As a consequence, distance measures based on cumulative distrbutions are usually utilized. Due to the fact that famous approaches cannot maintain uniqueness and symmetry in the multivariate case [10], the so-called *Localized Cumulative Distribution* (LCD) is introduced.

Definition 1 (Localized Cumulative Distribution, [10]) For a random vector $\underline{\boldsymbol{x}} \sim f(\underline{x}) : \mathbb{R}^N \rightarrow \mathbb{R}_+$, the LCD is defined by

$$F(\underline{m}, b) = \int_{\mathbb{R}^N} f(\underline{x}) \cdot K(\underline{x} - \underline{m}, b) \, d\underline{x} \ ,$$

with $F(\cdot, \cdot) : \Omega \rightarrow [0, 1]$, $\Omega \subset \mathbb{R}^N \times \mathbb{R}_+$. K is a symmetric and integrable kernel at position $\underline{m}$ and size b, with $K(\cdot, \cdot) : \Omega \rightarrow [0, 1]$. The kernel size b is considered to be equal in every dimension.

On basis of the Localized Cumulative Distribution, a modification of the Cramér-von Mises distance can be defined. The modified Cramér-von Mises distance defines a unique and symmetric distance for arbitrary multivariate probability distributions.

Definition 2 (Modified Cramér-von Mises Distance, [10]) The distance D between two probability densities $f_1(\underline{x}) : \mathbb{R}^N \rightarrow \mathbb{R}_+$ and $f_2(\underline{x}) : \mathbb{R}^N \rightarrow \mathbb{R}_+$ is defined by their corresponding LCDs $F_1(\underline{m}, b)$ and $F_2(\underline{m}, b)$ according to

$$D = \int_{\mathbb{R}_+} \omega(b) \int_{\mathbb{R}^N} (F_1(\underline{m}, b) - F_2(\underline{m}, b))^2 \, d\underline{m} \, db \ ,$$

where $\omega(b) : \mathbb{R}_+ \rightarrow [0, 1]$ defines a weighting function.

A deterministic discrete approximation of a given continuous probability density function can now be realized by minimizing the modified Cramér-von Mises distance by means of an optimization over the positions of all Diracs.

For more details about this deterministic density approximation technique, we refer the reader to [11].

IV. SNMPC WITH DETERMINISTIC DIRAC MIXTURE DENSITIES

In this section, we specify the operations for density propagation of the state estimate $\underline{\boldsymbol{x}}_k$ over time by using (3) and calculating the expected reward in (2), if the occurring densities are represented by deterministic Dirac mixture densities.

A. Propagation of the State Estimate

Let the probability density functions of the current state estimate and the system noise be given by Dirac mixture densities $\tilde{f}_k^x(\underline{x}_k)$ and $\tilde{f}_k^w(\underline{w}_k)$, respectively. Then, the joint probability density function of these two densities $\tilde{f}_k^z(\underline{z}_k) = \tilde{f}_k^x(\underline{x}_k) \cdot \tilde{f}_k^w(\underline{w}_k)$ can be written as $\tilde{f}_k^x(\underline{x}_k) \times \tilde{f}_k^w(\underline{w}_k)$, the Cartesian product of the Dirac components.

This leads to

$$\tilde{f}_{k+1}^p(\underline{x}_{k+1}) = \sum_{i=1}^{L_x \cdot L_w} \omega^{(i)} \cdot \delta(\underline{x} - a(\underline{z}_k^{(i)}, \underline{u}_k)) \ , \tag{5}$$

which is the analytical solution of (3) for Dirac mixture densities. Here, L_x and L_w denote the number of samples for $\tilde{f}_k^x(\underline{x}_k)$ and $\tilde{f}_k^w(\underline{w}_k)$.

As can be easily seen, (5) results in an increased complexity in run-time and storage. In detail, it leads to a multiplicative increase of Dirac components for each prediction step, namely $L_x \cdot L_w$. As a consequence, extrapolating a current state estimate over time results in an exponential increase of components. Thus, reduction of the complexity of representation is crucial for a feasible approach.

In this work, we employ a systematic technique for reducing the number of components that exploits independencies of $\tilde{f}_k^x(\underline{x}_k)$ and $\tilde{f}_k^w(\underline{w}_k)$ before propagating through $\underline{a}_k(\cdot, \cdot)$ as introduced in [12].

B. Calculating the Expected Reward

The expected one-step reward for a given estimate $\underline{\boldsymbol{x}}_k$ is defined by

$$E\left\{g(\underline{\boldsymbol{x}}_k, \underline{u}_k)\right\} = \int g(\underline{x}_k, \underline{u}_k) \cdot f_k^x(\underline{x}_k) d\underline{x}_k \ . \tag{6}$$

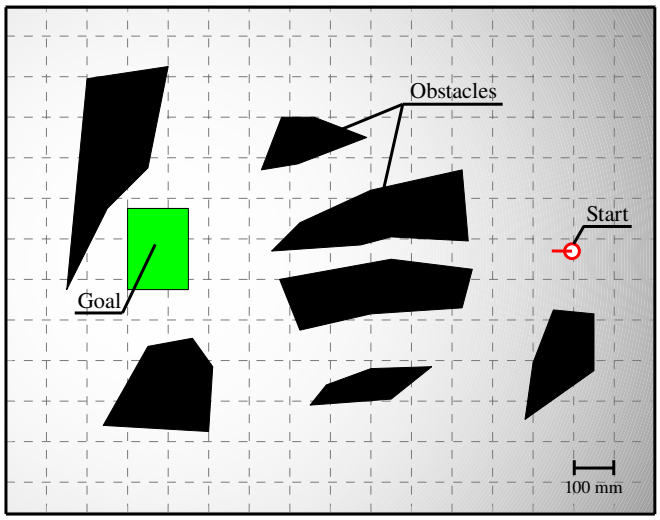


Fig. 2. This figure shows the setup for a designed scenario, where the red icon marks the initial position of a mobile robot. The green rectangle shows the goal region to be reached and the black polygons are obstacles to be avoided. Furthermore, the distance to the center of the goal region is rewarded illustrated by the light gray color gradient.

If $f_k^x(\underline{x}_k)$ is approximated by a Dirac mixture density, the insertion of (4) into (6) leads straightforward to

$$E\left\{g(\underline{\boldsymbol{x}}_k, \underline{u}_k)\right\} = \sum_{j=1}^{L} \omega^{(j)}\ g_k(\underline{x}_k^{(j)}, \underline{u}_k)\ , \tag{7}$$

i.e., the sum of individually evaluated component positions $\underline{x}_k^{(j)}$ with their corresponding weights $\omega^{(j)}$.

Considering (7), a discrete representation of an underlying continuous density offers two advantages: Firstly, no integration has to be performed and secondly, evaluation of an arbitrary reward function $g_k(\cdot)$ is possible, since the reward is only evaluated at specific positions.

C. Complexity Analysis

Considering the computational complexity, we would like to emphasize the fact that for a deterministic Dirac mixture density $\tilde{f}(\underline{x})$ with L Dirac components, (7) can be evaluated in $\mathcal{O}(L)$. Taking advantage of a small L becomes especially important, if the whole sequence of control inputs $\underline{u}_{k:k+H-1}$ (2) is taken into account. In this case, (7) has to be evaluated H times for every possible sequence of control inputs.

As for the prediction comprising the component reduction routine, (5) can be evaluated in polynomial time. For two N-dimensional deterministic Dirac mixture densities $\tilde{f}_k^x(\underline{x}_k)$ with L_x components and $\tilde{f}_k^w(\underline{w}_k)$ with L_w components, prediction can be performed in $\mathcal{O}(N \cdot (L_x^2 + L_w^2))$. For a more detailed complexity analysis concerning the prediction step, we refer the reader to [12].

V. SIMULATIONS

To demonstrate the benefits of using deterministic Dirac mixture densities in an SNMPC framework, the proposed approach was evaluated by means of a path planning scenario.

A. Simulation Setup

In particular, we consider a miniature robot introduced in [13] trying to reach a goal region without colliding with obstacles. This control objective is realized by applying the proposed SNMPC method. In order to model the desired behavior of the system, we have chosen the one-step reward function

$$g_k(\underline{x}_k, \underline{u}_k) = g_k^{(1)}(\underline{u}_k) + g_k^{(2)}(\underline{x}_k) + g_k^{(3)}(\underline{x}_k)\ ,$$

where $g_k^{(1)}(\underline{u}_k) = -1$ defines the motion cost and $g_k^{(2)}(\underline{x}_k) = 1/d(\underline{x}_k, \underline{c})$ represents a position reward given the L2 distance d between the robot position $\underline{x}_k$ and the goal center $\underline{c}$. Finally, the third component of the one-step reward function $g_k^{(3)}(\underline{x}_k)$ determines the position reward defined by

$$g_k^{(3)}(\underline{x}_k) = \begin{cases} 20, \text{ if true}(\text{“}\underline{x}_k \text{ in goal region”}) \\ -20, \text{ if true}(\text{“}\underline{x}_k \text{ in obstacle”}) \\ 0, \text{ otherwise} \end{cases} .$$

We use a system model that is similar to a two-wheeled differential drive and is given according to the nonlinear discrete-time system equation

$$\underline{\boldsymbol{x}}_{k+1} = \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_{k+1} \\ \boldsymbol{\phi}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_k + u_k^v \cdot \cos(\boldsymbol{\phi}_k + u_k^\phi) \\ \boldsymbol{y}_k + u_k^v \cdot \sin(\boldsymbol{\phi}_k + u_k^\phi) \\ \boldsymbol{\phi}_k + u_k^\phi \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_k^x \\ \boldsymbol{w}_k^y \\ \boldsymbol{w}_k^\phi \end{bmatrix} ,$$

where the system state $\underline{\boldsymbol{x}}_k = [\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{\phi}_k]^T$ defines the pose of the robot and the control input $\underline{u}_k = [u_k^v, u_k^\phi]^T$ consists of the step length u_k^v and the rotation angle u_k^ϕ.

Throughout the simulation, we used five possible control inputs

$$\mathcal{U} = \left\{ \begin{bmatrix} 50 \text{ mm} \\ 0 \end{bmatrix}, \begin{bmatrix} 50 \text{ mm} \\ \pm 10° \end{bmatrix}, \begin{bmatrix} 25 \text{ mm} \\ \pm 30° \end{bmatrix} \right\} .$$

Furthermore, the system noise $\underline{\boldsymbol{w}}_k = [\boldsymbol{w}_k^x, \boldsymbol{w}_k^y, \boldsymbol{w}_k^\phi]^T$ is assumed to be Gaussian white noise, where $[\boldsymbol{w}_k^x, \boldsymbol{w}_k^y]^T$ is concerning the position and $\boldsymbol{w}_k^\phi$ the rotation with the corresponding standard deviations $\sigma^x = \sigma^y = 4$ and $\sigma^\phi = 1°$, respectively.

In order to improve uncertain state estimation, we additionally consider distance measurements between the mobile robot and landmarks with fixed positions. Thus, the measurement equation is given by

$$\boldsymbol{z}_k = \sqrt{(\boldsymbol{x}_k - \check{x}_k)^2 + (\boldsymbol{y}_k - \check{y}_k)^2} + \boldsymbol{v}_k\ ,$$

where $[\check{x}, \check{y}]^T$ denotes the position of a landmark and $\boldsymbol{v}_k$ is the additive Gaussian white noise with the corresponding standard deviation $\sigma^v = 5$.

For state estimation, a standard particle filter with 500 particles is applied using sequential importance resampling [14]. In every time step k, the current state estimate represented by randomly drawn particles is approximated by a deterministic Dirac mixture density, in order to be able to perform the SNMPC operations with this kind of density representation.

The described control objective was applied in two scenarios using the proposed open-loop feedback SNMPC with a moving horizon of length $H = 4$.

TABLE I

SIMULATION RESULTS FOR THE HAND-CONSTRUCTED SCENARIO.

method (100 runs)	ø reward	ø steps	collisions
DM ($L = 12$)	-8.298	28.45	0
MC ($L = 12$)	-26.885	28.56	37
MC ($L = 144$)	-13.959	34.04	0

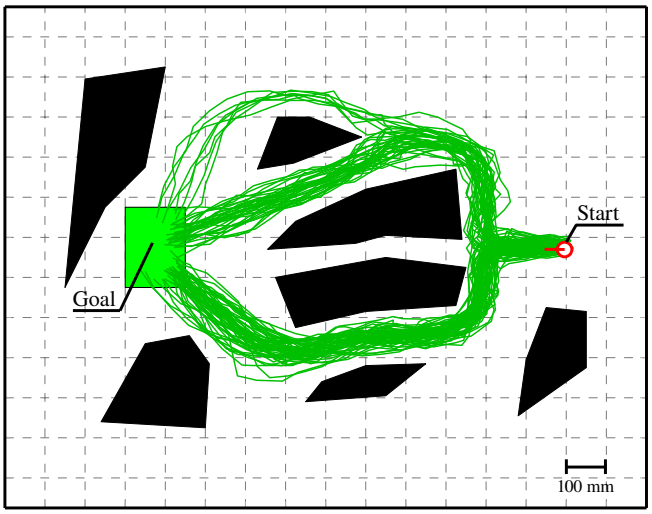
B. A Simple Hand-Contructed Scenario

In order to analyze the behavior of the proposed method, a simple hand-constructed scenario was used at first. In this scenario depicted in Fig. 2, the mobile robot can take different paths to the goal, which differ in length and distance to obstacles. The shortest path leads directly to the goal region. However, it is particularly narrow and thus, due to exogenous disturbances, uncertain state estimate, or an imprecise system model, the probability that the robot hits an obstacle is high, whenever the narrow path is chosen. Other more and less wide paths can be taken safely, taking into account that wider paths are longer.

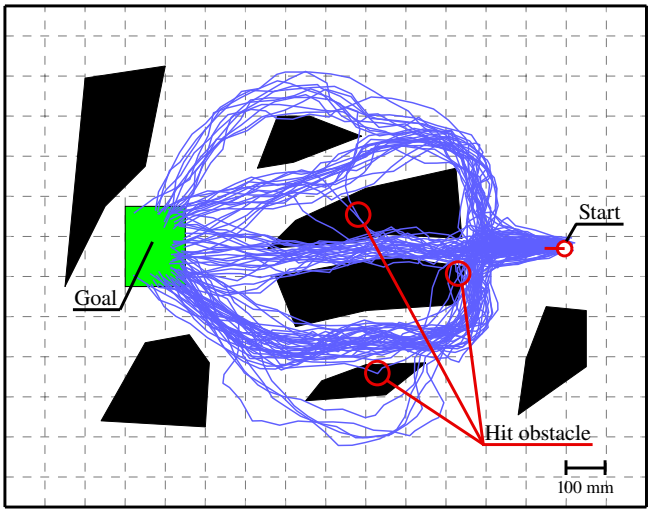
Fig. 3 (a) shows the trajectories of 100 Monte Carlo simulation runs of the proposed approach using $L = 12$ Dirac components. For comparison, a Monte Carlo sampling (MC) approach with 12 and 144 randomly drawn particles was evaluated by 100 Monte Carlo simulation runs. The results are shown in Fig. 3 (b) and Fig. 3 (c), respectively. As can easily be seen, the Monte Carlo approach with only 12 particles shows little robustness. More precisely, the narrow short path, which should be omitted because of uncertain disturbances, is often used and apparently, the robot often collides with obstacles. This is not surprising, since Monte Carlo sampling needs a certain number of particles to represent the original density well. In contrast, the Monte Carlo approach with 144 particles shows a very robust behavior and, thus, obstacles are omitted while longer paths are taken.

Applying the Dirac Mixture approach, robustness and efficiency are combined, since only 12 points of the reward function have to be evaluated. Furthermore, a greater constancy exists in the way paths are chosen, due to the deterministic procedure. While the Monte Carlo approach with 144 particles often chooses paths longer than necessary, the proposed approach uses short paths that are wide enough to be passed safely. Table I summarizes the simulation results of this scenario. In the absence of uncertanty and noise, a perfect run is accomplished by 20 steps, resulting in a slightly positive reward. Hence, disregarding the collision penalty, the average reward can be interpreted as an indicator of the taken detour, by means of the motion cost.

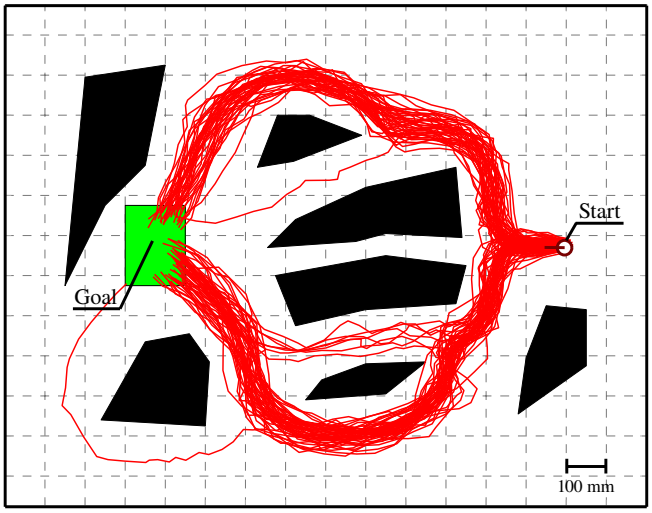
It is worth noticing that the proposed approach even surpasses the Monte Carlo sampling approach with $L = 12$ particles in terms of average steps taken to reach the goal, even though the short but narrow path is completely omitted. As one can already see from Fig. 3 (c), the MC approach with $L = 144$ particles takes longer paths and, thus, needs more steps on average.



(a) Deterministic Dirac mixture density with $L = 12$ components.



(b) Monte Carlo sampling with $L = 12$ particles.



(c) Monte Carlo sampling wih $L = 144$ particles.

Fig. 3. This illustration shows the true trajectories of 100 simulation runs, where the depicted results of subfigure (a) were calculated using the presented deterministic Dirac mixture density approach with $L = 12$ Dirac components. For comparison, the widely used Monte Carlo sampling approach with $L = 12$ particles (subfigure (b)) and $L = 144$ particles (subfigure (c)) were used.

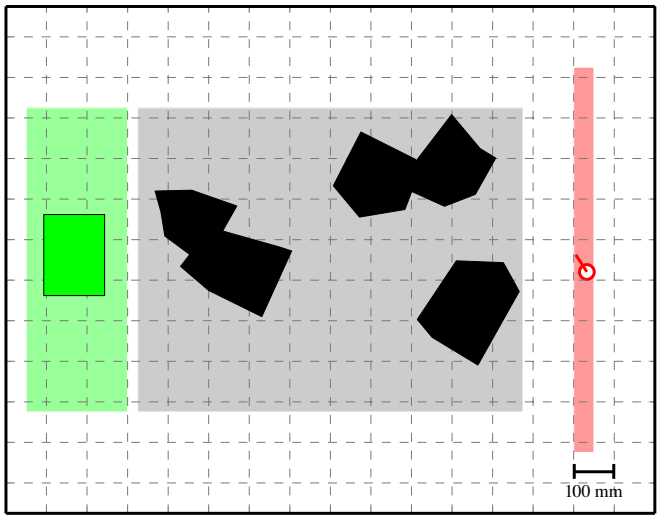


Fig. 4. This figure shows a randomly generated world, where the light green rectangle marks possible positions for the goal region and the light red rectangle for the starting position. Inside the grey rectangle obstacles are generated randomly. An exemplary scenario as described for Fig. 2 is shown.

TABLE II

SIMULATION RESULTS FOR THE RANDOMLY CONSTRUCTED SCENARIO.

method (200 runs)	ø reward	ø steps
DM ($L = 12$)	-10.91	30.68
MC ($L = 12$)	-13.50	30.38
MC ($L = 144$)	-12.53	32.51

The detours of all three approaches result in negative rewards, however the penalty taken by MC approach with $L = 12$ particles adds up to the worst average reward.

C. Performance in Several Randomly Constructed Scenarios

In order to prove the results in a more general manner, randomly generated worlds were also used for evaluating our approach. The generator is illustrated in Fig. 4, where goal region, starting position, and polygon shaped obstacles are being positioned randomly in specific predefined areas. The number of obstacles as well as the amount of edges for each obstacle is chosen randomly between 4 and 7. Furthermore, size and shape of each obstacle are random to a certain degree.

We conducted 200 Monte Carlo simualtion runs with one randomly generated world per run. The results are shown in Table II. As one can easily see, the average path length still shows similarities between both approaches with $L = 12$. The less frequent occurrence of long paths results in a decreased chance of hitting an obstacle for the Monte Carlo sampling approach and, therefore, in less penalty. Due to the fact that most of the generated worlds are simpler than the constant scenario, the difference of the average path length is decreased. However, the main statement is confirmed.

VI. CONCLUSIONS

In this paper, the integration of a discrete density representation, in which the samples are determined systematically by minimizing a distance measure, in an SNMPC framework is presented. Compared to random sampling, fewer samples are necessary to represent the underlying probability density function while achieving the same approximation quality. This property is particularly advantageous, if the evaluation of the expected reward is computationally expensive, e.g., because of a complex reward function or system model. By means of a path planning problem of a mobile robot, we have emphasized the benefits of our proposed approach.

VII. ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the support of Toshiyuki Ohtsuka for helpful discussions about the ideas of this paper.

REFERENCES

[1] S. Qin and T. Badgwell, "An Overview of Industrial Model Predictive Control Technology," in *Fifth International Conference on Chemical Process Control*, vol. 93, no. 2, 1997, pp. 232–256.

[2] F. Weissel, M. F. Huber, and U. D. Hanebeck, "A Closed–Form Model Predictive Control Framework for Nonlinear Noise–Corrupted Systems," in *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, vol. SPSMC, Angers, France, May 2007, pp. 62–69.

[3] D. Nikovski and M. Brand, "Non-linear Stochastic Control in Continuous State Spaces by Exact Integration in Bellman's Equations," in *13th International Conference on Automatic Planning & Scheduling (ICAPS 2003)*, pp. 91–95.

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

[5] L. Blackmore, "A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[6] G. Hoffmann and C. Tomlin, "Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, 2010.

[7] S. Thrun, "Monte Carlo POMDPs," *Advances in neural information processing systems*, vol. 12, pp. 1064–1070, 2000.

[8] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[9] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, Random Variables, and Stochastic Processes*. McGraw-hill New York, 1965, vol. 196.

[10] 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 (MFI 2008)*, Seoul, Republic of Korea, Aug. 2008, pp. 33–39.

[11] U. D. Hanebeck, M. F. Huber, and V. Klumpp, "Dirac Mixture Approximation of Multivariate Gaussian Densities," in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.

[12] H. Eberhardt, V. Klumpp, and U. D. Hanebeck, "Optimal Dirac Approximation by Exploiting Independencies," in *Proceedings of the 2010 American Control Conference (ACC 2010)*, Baltimore, Maryland, Jun. 2010.

[13] F. Weissel, M. F. Huber, and U. D. Hanebeck, "Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, California, Nov. 2007, pp. 2474–2479.

[14] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.