Stochastic Nonlinear Model Predictive Control Based on Deterministic Scenario Generation

Christof Chlebek and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
Christof.Chlebek@kit.edu, Uwe.Hanebeck@ieee.org

***Abstract*—We consider closed-loop feedback (CLF) stochastic model predictive control of nonlinear time-invariant systems with imperfect state information. In this class of control problems, future information feedback is considered in the decision making process, and thus, the effect of the control influencing the state uncertainty is taken into account. The main challenge in the solution is to find a good approximation to the arising stochastic dynamic programming problem, which is computationally not tractable. In this work, future information is considered in the form of conditional state probability densities. Thus, the objective is it to optimize the state and its uncertainty as a combined problem. We propose to discretize the state space by a novel scenario generation approach based on deterministic sampling. A distance based threshold determines the narrowness of the discretization. The dynamic programming problem is formulated such that the approximate cumulative control cost function can be explicitly evaluated offline. The online calculation consists of a one-step prediction and the interpolation of the explicit cost function in order to calculate the control input. The effectiveness of this novel method is presented by means of a simulation.**

I. Introduction

The main target of stochastic model predictive control is to optimize the behavior of a system based on a stochastic model by minimizing a cost function. In particular, this work addresses closed-loop feedback (CLF) stochastic model predictive control, where the controller calculates a control policy that incorporates information about all possible future feedback information. In the considered stochastic framework, we assume no perfect state information, but feedback is only provided through disturbed measurements of the system state. In literature, this is referred to as the imperfect state information [1] or partial observability [2]. In CLF control, state estimation is integrated into the control problem, where an optimal control policy implicitly optimizes the state and its uncertainty together. This problem formulation leads to the optimization of functions, in order to calculate the optimal policy [1]. CLF control is solvable exactly only for special cases, such as the linear-quadratic-Gaussian (LQG) control problem or systems with a finite number of states and inputs. We consider general nonlinear systems, where in most cases control has dual effect. This means that in addition to the system state, the control also affects the state uncertainty and thus, the state estimation [3]. While in cases ignoring the uncertainty and assuming certainty equivalence (CE) yields a good approximation of the optimal CLF controller, in others a CE controller might be not capable of controlling a system due to insufficient information gain. Approaches to solving the problem of insufficient measurement information range from application of sophisticated and possibly expensive sensors (e.g., estimation of tire-road friction coefficient [4]) to using human-in-the-loop approaches in unfavorable sensing environments (e.g., control of irrigation canals [5]). On the other hand, constrains on production cost and technical restrictions motivate novel control approaches capable of handling the dual effect. In the following, we give a quick overview over the current literature in this area.

Explicit dual control uses a modification of the cost function. The original control objective is calculated by a point estimate and extended by adding a cost penalizing the uncertainty explicitly. Not considering uncertainty as part of the control objective may lead to an inappropriate weighting of the uncertainty, since it is not considered in the environment of the original control objective [6].

Assuming only discrete finite sets of inputs and measurements, the optimal closed-loop policy can be calculated by generating a search tree by means of all combinations of inputs and measurements over the control horizon. This approach has an exponential growth over the control horizon and thus, is limited to a short horizon [7].

This work is also inspired by random scenario generation used in control optimization [8], [9], [10]. This approach has gained much attention in the recent years, since bounds on the number of required samples for a guaranteed quality have been established [8]. Unfortunately these approaches assume systems with direct accessibility of the state, and thus cannot be used for the considered system class. In order to reduce the amount of resulting scenarios offline, the Wasserstein distance can be utilized to delete the scenario most likely to be covered by other scenarios. This can be repeated until a desired amount of scenarios has been deleted [11]. A different approach is to use vector quantization, where the scenarios are represented as part of a code book. Reduction is achieved by consolidation of state probability densities in Voronoi cells [12].

Originally considering discrete state spaces, and thus, historically differently motivated, POMDPs (partially observable Markov decision processes) have been extended to continuous state spaces in recent years [2], [13], [14]. In this field of research, the space of estimated states is called belief space and the cost function is explicitly computed offline by reinforcement learning methods. The belief space can be discretized by Monte Carlo exploration [15] or discretization of parametric density representations [13], [14]. Furthermore, Monte Carlo methods used in the online estimation and parametric densities representations of the belief space can be combined using the Kullback-Leibler divergence [16].

A. Contributions

The calculation of a CLF policy involves the evaluation of a cumulative cost function. The key idea of the presented work is to apply deterministic scenario generation in contrast to Monte Carlo approaches. This ensures a good coverage of the reachable state space. Utilization of a deterministic sampling procedure allows for the consideration of arbitrary probability densities underlying the system. A pruning approach based on the Wasserstein distance, preserves only scenarios with sufficient dissimilarity and thereby, considerably reduces the complexity and renders the problem tractable. Besides using the same distance as in [11], the presented work utilizes pruning during the scenario generation, instead of reducing the number of scenarios afterwards. As a complement to the new scenario generation approach, we introduce a novel calculation procedure for the online system control. The optimization is performed by the minimization of approximated cost, in terms of a one-step prediction and interpolation of predicted states picked offline by the scenario generation.

B. Outline of this Paper

The remainder of this work is structured in a top down fashion. First, in the following section, the problem is stated formally by introducing the considered system class, the closed-loop control problem, and assumptions made. In Sec. III, we formulate the online optimization of the closed-loop optimal control input by means of the optimization of a one-step prediction. This can only be computed given the cumulative cost function for the future cost is known, which gives rise to Sec. IV. Here, we present a method for the approximation of future cost evaluated for a finite set of predicted states. Sec. V introduces the scenario generation approach and give details on the offline calculation of the cumulative cost function. Finally, in Sec. VI, the presented approach is evaluated and Sec. VII concludes this work.

II. Problem Formulation

In this section, the formal problem is stated in three parts. First, the considered system class of nonlinear stochastic discrete-time measurement feedback systems is introduced. Second, the general closed-loop stochastic model predictive problem is stated. Finally third, the general problem is relaxed by assuming that information used by the controller can be sufficiently represented by a conditional state probability density.

A. Considered System

Let us denote k as the current time step of a discrete-time setup. We consider stochastic nonlinear systems of the form

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

where the continuous-valued random vector $\underline{\boldsymbol{x}}_k \in \mathcal{X}$ is the system state characterized by the probability density $f_k^x(\underline{x}_k)$. The probability density $f_0^x(\underline{x}_0)$ characterizing the initial state $\underline{\boldsymbol{x}}_0$ is assumed to be known. The also continuous-valued control input vector $\underline{u}_k \in \mathcal{U}$ is chosen from the bounded set $[\underline{u}_k^{min}, \underline{u}_k^{max}]$. The state is propagated over time by the system function $a : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ and is affected by i.i.d. noise $\underline{\boldsymbol{w}}_k \in \mathcal{W}$, with $\underline{\boldsymbol{w}}_k \sim f_k^w(\underline{w}_k)$.

The system state $\underline{\boldsymbol{x}}_k$ is not directly accessible, but can be measured by the nonlinear measurement function

$$\underline{\boldsymbol{y}}_k = h(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{v}}_k) \ , \tag{2}$$

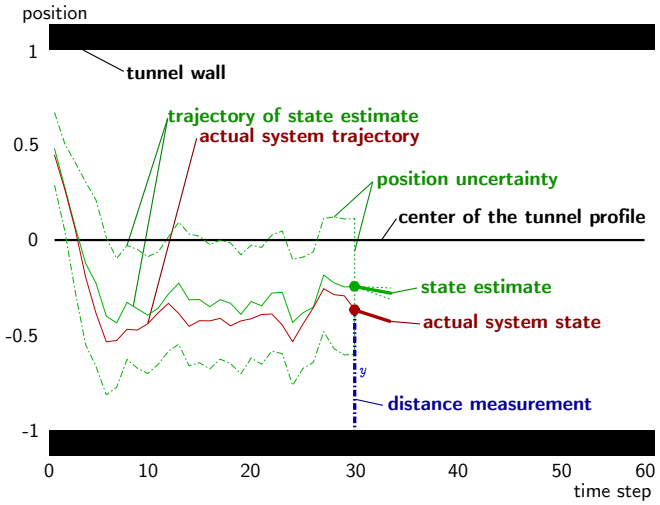


Fig. 1. Visualization of the considered example system. The figure shows the actual system state (red), its corresponding state estimate (green), and a distance measurement (blue).

where $\underline{\boldsymbol{y}}_k \in \mathcal{Y}$ is the measurement output characterized by $f_k^y(\underline{y}_k)$ and $h : \mathcal{X} \times \mathcal{V} \to \mathcal{Y}$ denotes the measurement function. The measurement itself is disturbed by the arbitrary white noise $\underline{\boldsymbol{v}}_k \in \mathcal{V}$, with $\underline{\boldsymbol{v}}_k \sim f_k^v(\underline{v}_k)$, which is also state-independent.

Example System: Autonomous Vehicle in a Tunnel

Let us visualize this system class by a simple example. We assume an autonomous car driving through a straight tunnel with constant speed. We are only interested in the cars position relative to the walls; thus, we can describe the system by the two-dimensional model

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

where the system state $\underline{\boldsymbol{x}}_k = [\boldsymbol{x}_k, \boldsymbol{\phi}_k]^T$ is composed of the position, which is given relative to the center of the tunnel profile, and the orientation. The two-dimensional noise vector is additively disturbing the translational and the rotatory part of the state, respectively. The control input u_k is the steering input, changing the orientation. The car is equipped with a distance sensor measuring the closest distance to an obstacle, i.e., one of the two tunnel walls. The sensor model given by

$$y_k = x_{wall} - |\boldsymbol{x}_k| + \boldsymbol{v}_k$$

where x_{wall} denotes the distance of the walls relative to the center of the tunnel profile and $\boldsymbol{v}_k$ is an additive noise term. This example system is depicted in Fig. 1.

B. Closed-Loop Control

We consider closed-loop control policies, which take into account that future decisions will be based on more available information. This generally leads to the optimization of functions mapping information to control inputs. Let us denote the available information at time step k as

$$\begin{aligned} I_0 &= \{f_0^x\} \ , \\ I_k &= \{f_0^x, \underline{y}_1, \underline{y}_2, \ldots, \underline{y}_k, \underline{u}_0, \underline{u}_1, \ldots, \underline{u}_{k-1}\} \ , \end{aligned}$$

for $k = 1, 2, \ldots$, where I_k is called the information set. An admissible policy π consists of a sequence of functions

$\pi = \{\mu_0, \mu_1, \ldots\}$, where each function μ_k maps the available information I_k to a system input $\underline{u}_k$. The objective is to find an optimal policy π^*, which minimizes the expected cumulative cost function

$$J^\pi = \mathop{\mathbf{E}}_{\substack{\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{w}}_k, \underline{\boldsymbol{v}}_k \\ k=0,\ldots,N-1}} \left\{ \gamma^N \tilde{g}(\underline{\boldsymbol{x}}_N) + \sum_{k=0}^{N-1} \gamma^k g(\underline{\boldsymbol{x}}_k, \mu_k(I_k)) \right\} , \tag{3}$$

where $g : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^+$ denotes the application-specific one-step cost mapping every state-input pair to a real number and $\tilde{g} : \mathcal{X} \to \mathbb{R}$ is the terminal cost. The control horizon is denoted by N and $\gamma \in (0, 1]$ is the step-discounting factor. If the horizon N is chosen as $N \to \infty$, the discounting factor should be $\gamma < 1$. Thus, we are looking for π^* such that,

$$J^{\pi^*} = \min_{\pi \in \Pi} J^\pi ,$$

where Π is the set of all admissible policies.

Using Bellman's principle of optimality, the minimal cost J^{π^*} and consequently the closed-loop optimal input sequence $U^* = (u_0^*, u_1^*, \ldots u_{N-1}^*)$, can be obtained by

$$\begin{aligned} J^{\pi^*} = \min_{u_0} \mathbf{E}\Big\{ & g(\underline{\boldsymbol{x}}_0, \underline{u}_0) + \gamma \min_{u_1} \mathbf{E}\Big\{ g(\underline{\boldsymbol{x}}_1, \underline{u}_1) + \gamma^2 \min_{u_2} \ldots \\ & \ldots + \gamma^{N-1} \min_{u_{N-1}} \mathbf{E}\{ g(\underline{\boldsymbol{x}}_{N-1}, \underline{u}_{N-1}) + \gamma^N \tilde{g}(\underline{\boldsymbol{x}}_N) | I_{N-1} \} \ldots \\ & \ldots | I_2 \} | I_1 \Big\} | I_0 \Big\} , \end{aligned}$$

which are nested optimization problems minimizing the expected cost conditioned on available information at each time step. Thus, the problem can only be solved by starting the calculation from the innermost expectation. This can be restated in the backward-recursive formulation based on the concept of dynamic programming, given by

$$\begin{aligned} J_{N-1}^{\pi^*}(I_{N-1}) &= \min_{u_{N-1}} \mathbf{E}\Big\{ g(\underline{\boldsymbol{x}}_{N-1}, \underline{u}_{N-1}) + \gamma \tilde{g}(\underline{\boldsymbol{x}}_N) | I_{N-1} \Big\} , \\ J_k^{\pi^*}(I_k) &= \min_{u_k} \mathbf{E}\Big\{ g_k(\underline{\boldsymbol{x}}_k, \underline{u}_k) + \gamma J_{k+1}^{\pi^*}(I_{k+1}) | I_k \Big\} . \end{aligned} \tag{4}$$

In favor of a better readability, we have omitted the random variables $\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{w}}_k, \underline{\boldsymbol{v}}_k$ in the expectation operators above. By performing this calculation, we get the optimal policy, and thus, can generate the closed-loop optimal control inputs. Let us elaborate the explanation why closed-loop optimal control is so important on the introduced example.

Example System: Autonomous Vehicle in a Tunnel

The control problem is to keep the car safe from crashing into a wall. Hence, the cost function is minimal if the vehicle is farthest from the walls (i.e., closest the center of the tunnel profile). In this scenario, the one-step cost function is given by the quadratic cost

$$g(\underline{x}_k, u_k) = \underline{x}_k^T \mathbf{Q} \underline{x}_k + u_k^T \mathbf{R} u_k ,$$

where $\underline{x}_k = [x_k, \phi_k]^T$ is the state, u_k denotes the control input as given in the previous section, and $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices.

Having established the objective in terms of the cost function, it is easy to see that the cost optimal position for one step is $\underline{x}_k = [0, 0]^T$. In order to see, why this is not the optimal position to actually hold onto, we need to examine the sensor model, which has been illustrated in Fig. 2. The sensor model

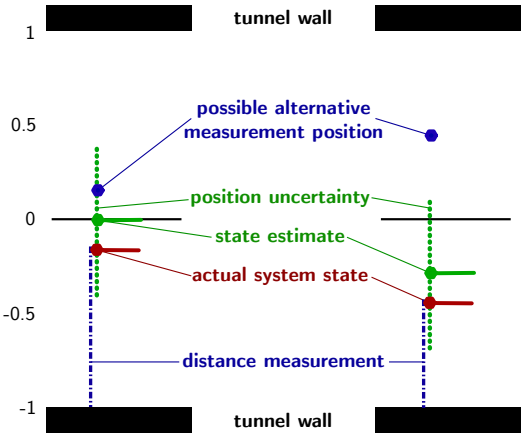


Fig. 2. Illustration of the observability problems in the considered example scenario. Positioning the mean of the state estimate on the cost minimal tunnel center results in no information update of the mean (left). Controlling the state estimate slightly off the center brings about a change of this situation (right).

is a function symmetric around the center of the tunnel profile, i.e., the previously established cost optimal position. Thus, having an information update showing a deviation from the tunnel center (i.e., $y \neq x_{wall}$), we have no information about the direction of the deviation. Thus, the information update is equally probably on both sides and the mean of the estimation stays unchanged. This problem can be avoided by holding a position off the center, which brings about the optimal control policy.

C. Structural Assumption

In general, the information set grows over time, as more and more information becomes available. In this work, we assume that the controller has no direct access to the measurement feedback $\underline{y}_k$. A decision can only by based on the conditional state probability density $\underline{\boldsymbol{x}}_k^e \sim f_k^x(\underline{x}_k | I_k)$, as a representation of the information set. The assumed system structure is illustrated in Fig. 3. We label the random-vector by the superscript e as in estimated state. The density is recursively calculated using Bayes' law according to

$$f_k^x(\underline{x}_k | I_k) = \frac{f_k^L(\hat{\underline{y}}_k | \underline{x}_k) \cdot f_k^x(\underline{x}_k | I_{k-1}, \underline{u}_{k-1})}{\int f_k^L(\hat{\underline{y}}_k | \underline{x}_k) \cdot f_k^x(\underline{x}_k | I_{k-1}, \underline{u}_{k-1}) \, \mathrm{d}\underline{x}_k} ,$$

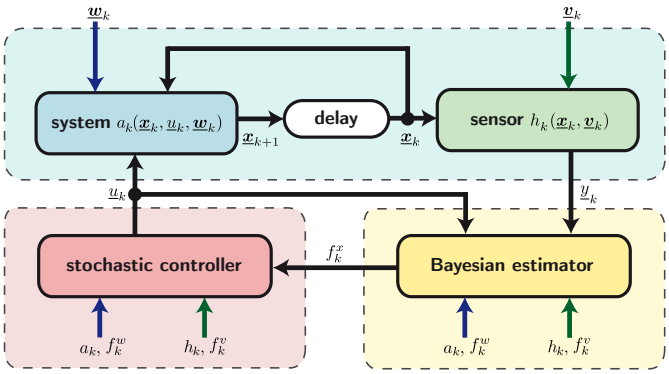


Fig. 3. The system structure considered in this work. The controller has not direct access to the measurement feedback, but has to compute decisions based on the conditional state probability density.

where $f_k^L(\hat{\underline{y}}_k|\underline{x}_k)$ is the *likelihood function* for a given measurement $\hat{\underline{y}}_k$. It is derived from the conditional probability density

$$f_k^y(\underline{y}_k|\underline{x}_k) = \int \delta(\underline{y}_k - h(\underline{x}_k, \underline{v}_k)) \cdot f_k^v(\underline{v}_k)\ \mathrm{d}\underline{v}_k\ ,$$

which characterizes the probability of occurrence for all possible measurements $\underline{\boldsymbol{y}}_k$ given the measurement model (2) and the prior $\underline{\boldsymbol{x}}_k^p \sim f_k^x(\underline{x}_k|I_{k-1}, \underline{u}_{k-1})$. We marked the prior by the superscript p as in predicted state. The probability density $f_k^x(\underline{x}_k|I_{k-1}, \underline{u}_{k-1})$ is calculated by the Chapman-Kolmogorov equation according to

$$f_k^x(\underline{x}_k|I_{k-1}, \underline{u}_{k-1}) = \int f_k^T(\underline{x}_k|\underline{x}_{k-1}) \cdot f_k^x(\underline{x}_{k-1}|I_{k-1})\ \mathrm{d}\underline{x}_k\ ,$$

where $f_k^T(\underline{x}_k|\underline{x}_{k-1})$ is the state transition density for the system model (1), given by

$$f_k^T(\underline{x}_k|\underline{x}_{k-1}) = \int \delta(\underline{x}_k - a(\underline{x}_{k-1}, \underline{u}_{k-1}, \underline{w}_{k-1})) \cdot f_{k-1}^w(\underline{w}_{k-1})\ \mathrm{d}\underline{w}_{k-1}\ .$$

The assumption that the information set can be represented by a conditional state probability density implies a priori separation of state estimation and control. Although an implicit consideration of the estimation cost in the control loop is preserved, we would like to emphasize that this is a special case of closed-loop optimal feedback control, where the calculation of the cost function (4) can be formulated as

$$J_k^{\pi^*}(I_k) = J_k^{\pi^*}(\underline{\boldsymbol{x}}_k^e)\ . \tag{5}$$

This formulation is only optimal under the assumption that the state estimate is a sufficient statistic for the information set.

III. Separation of Prediction and Estimation Costs

For the considered time-invariant models (1) and (2), the time-invariant cost function (3), and the sufficiency assumption (5), the optimal admissible policy can be reduced to a stationary policy of the form $\pi^* = \{\mu^*, \mu^*, \ldots\}$, where μ^* is now mapping estimated states to the optimal system input, i.e., $\mu^*(\underline{\boldsymbol{x}}_k^e) = \underline{u}_k^*$. Consequently, we denote the time-invariant optimal cost as $J^{\pi^*}(\underline{\boldsymbol{x}}_k)$.

In order to specify the function $\mu^*(\underline{\boldsymbol{x}}_k^e)$, we first subdivide $J^{\pi^*}(\underline{\boldsymbol{x}}_k^e)$ into two parts. For now, let us consider the cost of predicted states $\underline{\boldsymbol{x}}_k^p$. Since, we cannot assume any information about the measurement, the expected cost for $\underline{\boldsymbol{x}}_k^p$ is calculated over the expectation of all possible measurements by

$$J^{\pi^*}(\underline{\boldsymbol{x}}_k^p) = \mathop{\mathbf{E}}_{\underline{\boldsymbol{y}}_k}\left\{ J^{\pi^*}(\underline{\boldsymbol{x}}_k^e)|\underline{\boldsymbol{x}}_k^p \right\}\ . \tag{6}$$

Using this formulation, we can now calculate the cost of an estimated state $J^{\pi^*}(\underline{\boldsymbol{x}}_k^e)$ recursively, subject to the predicted future state, by

$$J^{\pi^*}(\underline{\boldsymbol{x}}_k^e) = \min_{\underline{u}_k} \mathop{\mathbf{E}}_{\underline{\boldsymbol{x}}_k^e}\left\{ g(\underline{\boldsymbol{x}}_k^e, \underline{u}_k) + \gamma J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)|\underline{\boldsymbol{x}}_k^e, \underline{u}_k \right\}\ . \tag{7}$$

In principle, this means that if we know the future cost for all reachable predicted states of a given estimated state, we can calculate the closed-loop optimal system input $\underline{u}_k$ by optimizing over a one-step prediction evaluating (7). Thus, knowledge of future measurements is unnecessary as long as we can evaluate the cost function (6). We will exploit this fact, by an explicit representation of (6), which can be calculated offline and compute the optimal state-to-input mapping function by

$$\mu^*(\underline{\boldsymbol{x}}_k^e) = \arg\min_{\underline{u}_k} \mathop{\mathbf{E}}_{\underline{\boldsymbol{x}}_k^e}\left\{ g(\underline{\boldsymbol{x}}_k^e, \underline{u}_k) + \gamma J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)|\underline{\boldsymbol{x}}_k^e, \underline{u}_k \right\}\ , \tag{8}$$

where the main assumption is that $J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$ is known.

IV. Approximation of Future Cost

The evaluation of $\mu^*(\underline{\boldsymbol{x}}_k^e)$, as introduced in the previous section, is unfortunately computationally not tractable. Although, the estimated state, in theory, captures all information relevant to the controller accumulated up to every time step k for many systems, this calculation can only be performed in special cases. In general, the resulting state densities can take arbitrary shapes and therefore, are not easily represented. In recent years, many filtering techniques have been developed in order to cope with this problem and render the filtering problem tractable through different approximations. In general, arbitrary probability densities can be meaningfully approximated by Dirac mixture densities of the form

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

where $\underline{x}^{(j)}$, for $j = 1, ..., L$, denotes the position of the jth Dirac component with the corresponding positive weight $\omega^{(j)}$, with $\sum_{j=1}^{L} \omega^{(j)} = 1$. In this work, we assume the weight to be equal for every Dirac and thus, $\omega^{(j)} = 1/L$. This is a widely used representation for probability density functions and is utilized in many different Bayesian filter approaches, to name the least: The Unscented Kalman Filter (UKF) uses Diracs (or sigma points, as they are often called in this context) for statistical linearization [17]. A similar approach is taken by the Smart Sampling Kalman Filter (S2KF), which uses more Diracs for approximation and thus achieves a better quality [18]. The Progressive Gaussian Filter 42 (PGF42) uses a progressive approach to update sample positions and performs well with strong nonlinearities [19]. Finally, the particle filter is a good approach for multi-modal densities, but needs a large number of samples [20].

Besides the calculation of the estimated state, assumptions about the cumulative cost function have to be satisfied. For a continuous-valued state space $\mathcal{X}$ with noise $\underline{\boldsymbol{w}}_k$ affecting the state and a continuous-valued system input $\underline{u}_k \in \mathcal{U}$, there are infinitely many reachable predicted states $\underline{\boldsymbol{x}}_{k+1}^p$. Furthermore, $J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$ has to be evaluated for infinitely many future possible measurements in order to calculate the expected cost. Thus, the assumption that $J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$ cannot be satisfied in this general formulation.

In the first instance, let us assume that $J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$ is not given everywhere, but only its value at a finite set of representative predicted states to be known. We introduce an approximation $\tilde{J}(\underline{\boldsymbol{x}}_{k+1}^p) \approx J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$, where we use known values close to the predicted $\underline{\boldsymbol{x}}_{k+1}^p$. We propose to use the Wasserstein distance, which we only consider for the here relevant Dirac mixture densities.

Definition 1 (Wasserstein distance) [21] Let us consider two random variables $\underline{\boldsymbol{x}}_s$ and $\underline{\boldsymbol{x}}_t$ characterized by the equally

weighted Dirac mixture densities $f_s(\underline{x}_s) = \sum_{i=1}^{M} \frac{1}{M}\delta(\underline{x} - \underline{s}_i)$ and $f_t(\underline{x}_t) = \sum_{j=1}^{M} \frac{1}{M}\delta(\underline{x} - \underline{t}_j)$, respectively, where the positions of the Dirac components are given by the two sets $S = \{\underline{s}_1, \ldots, \underline{s}_M\}$ and $T = \{\underline{t}_1, \ldots, \underline{t}_M\}$. The Wasserstein distance for Dirac mixture densities is calculated by

$$D(\underline{\boldsymbol{x}}_s, \underline{\boldsymbol{x}}_t) = \left(\frac{1}{M} \inf_{\lambda \in \Lambda_M} \left(\sum_{i=1}^{M} d\left(\underline{s}_i, \underline{t}_{\lambda(i)} \right)^q \right) \right)^{1/q} , \quad (9)$$

where the infimum is computed over all permutations Λ_M of the set $\{1, \ldots, M\}$. The notation $\underline{t}_{\lambda(i)}$ is the i-th element of the permutation λ of T.

The computation of the optimal permutation can be efficiently implemented using the Hungarian algorithm [22]. In this work, we use the 2nd order Wasserstein distance and the Euclidean distance as metric $d(\cdot, \cdot)$.

Having established the distance between two Dirac mixture densities, we evaluate $\tilde{J}(\underline{\boldsymbol{x}}_{k+1}^p)$ by finding the k-nearest neighbor of $\underline{\boldsymbol{x}}_{k+1}^p$, where the cost function is known. We will use n neighbors, which will be denoted by $\underline{\boldsymbol{x}}^{p,1}, \ldots \underline{\boldsymbol{x}}^{p,n}$ and their calculated Wasserstein distances to $\underline{\boldsymbol{x}}_{k+1}^p$ will be denoted by $d_1, \ldots d_n$, respectively. The approximation can be calculated by a weighted sum of pre-calculated values of the cost function, i.e.,

$$\tilde{J}(\underline{\boldsymbol{x}}_{k+1}^p) \approx \sum_{i=1}^{n} \omega_i \cdot J^{\pi^*}(\underline{\boldsymbol{x}}^{p,i}), \quad (10)$$

where ω_i are the weights consisting of the normalized distances $\omega_i = d_i / \sum_{i=1}^{n} d_1 + \ldots + d_n$. With this approximation we can calculate the state-to-input function (8) and thus, approximate the optimal policy.

V. Explicit Calculation of the Cost Function

Until now, we have assumed that we already have an explicitly calculated cost function $J^{\pi^*}(\underline{\boldsymbol{x}}_{k+1}^p)$, comprising the cumulative cost of all future time steps. In this section, we want to concentrate on the discretization of this cost function by a finite set of densities and the computation of their corresponding cost.

A. Deterministic Scenario Generation

In order to discretize the cost function, we use a novel scenario generation approach based on a deterministic approximation of the noise probability density distributions. The computational cost is highly dependent on the amount of realizations. Thus, we want to approximate the original distribution as roughly as possible, while still maintaining a representative set of samples. This is done by minimizing a distance measure called Localized Cumulative Distribution (LCD) distance [23], which is a kernel based distance for arbitrary probability densities with equal means. In the original work, box kernels are used [23]. Later works consider Gaussian type kernels [24], [25]. Computing an equal weighted Dirac mixture density as an approximation of an arbitrary probability density we minimize the distance between both densities by means of the optimal position of the Dirac components. This gives rise to a deterministic discrete approximation, where the number of Diracs can be chosen arbitrarily. For approximations of Gaussian densities, the gradient of the distance used in the optimization, can be calculated in closed form [25].

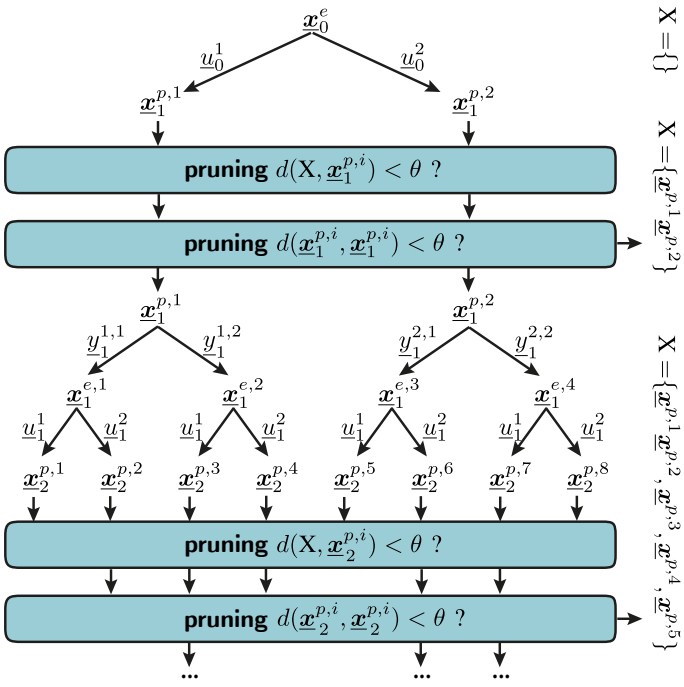


Fig. 4. Illustration of the deterministic scenario generation approach, which exploits the similarity between states and thereby, implementing an effective pruning method.

Having sampled the noise, we finally choose a finite set of possible starting positions and use a discrete set of inputs to emulate the system. The emulation is done for both, the actual system and the estimation, just like the actual system illustrated by Fig. 3. Later although, only the probability densities, resulting from the estimation, have to be saved. The emulation of the system serves for the generation of meaningful measurements for the estimation. The emulation can be either performed until the complete state space has been covered, or for an arbitrary but sufficiently long time horizon.

B. Effective Pruning for Scenario Generation

Branching several possible outcomes each time step in every path of the presented scenario generation approach, leads to a computationally highly demanding problem. An exhaustive calculation implicates a tree structure, where the amount of calculations is growing exponentially. Hence, the presented approach relies on an effective pruning algorithm, which allows the scenario generation to cover the relevant state space consistently, and at the same time prevent an excessive calculation. The main idea is to define a saturation value, which indicates a sufficiently dense coverage of the state space. For this saturation value, we define a threshold θ, which indicates the minimal distance between two explicitly calculated predicted states of the cost function. We again use the Wasserstein distance (9) and thus, we require

$$D(\underline{\boldsymbol{x}}^{p,i}, \underline{\boldsymbol{x}}^{p,j}) > \theta \ , \quad (11)$$

for all $i \neq j$. In the following, we will describe a pruning procedure exploiting θ, in the sense that the possibly largest amount of paths are preserved, and paths close in state space, i.e, violating requirement (11), are combined by discarding states.

Let us denote $X = \{\underline{\boldsymbol{x}}^{p,1}, \underline{\boldsymbol{x}}^{p,2} \ldots\}$ as the set of predicted states saved for the approximation of the cost function (10). For each possible pair of elements of X the requirement (11) holds. We predict one step of every path using the scenario generation described in the previous section. This can be interpreted as a breadth-first approach in the tree-structure, which leads to a new set of predicted states $\tilde{X} = \{\tilde{\underline{\boldsymbol{x}}}^{p,1}, \tilde{\underline{\boldsymbol{x}}}^{p,2} \ldots\}$ potentially to be integrated into X. The optimal procedure, as stated above, is to find a subset of $\tilde{X}$ with the largest amount of elements.

First, we discard all elements of $\tilde{X}$, which violate (11) with at least one element of X. Second, for the remaining elements, we calculate the distance between each other. Finally, we add the largest subset of states $\underline{\boldsymbol{x}}^{p,i}$ to $\tilde{X}$ that satisfy (11). This can be interpreted as a graph problem, where elements are nodes that are connected, if they are mutually exclusive. Thus, we want to disconnect a graph by deleting nodes, while maintaining as many nodes as possible. This is equivalent to the deletion of the minimal vertex cover of the graph, which to find is NP-complete [26]. As approximation to the optimal solution, we propose to use a small modification of the factor two greedy approximation, where successively an edge is chosen and both nodes connected are deleted. Instead of choosing an arbitrary edge and deleting both nodes, we first choose the node with the highest connectivity (i.e., the most connecting edges). Then we choose from its connected nodes again the one with the highest connectivity. If this partner has a larger connectivity than one (i.e., is connect still with other nodes), we delete both nodes, otherwise we delete only the first. This is done successively, until the graph is fully disconnected. The residual nodes are the new states to be subjoined to X.

Starting with only the newly added states, the next step of the scenario generation is performed. Paths of discarded states are therefore pruned from the scenario generation procedure. We have illustrated this procedure in Fig. 4 and show an exemplary outcome of the scenario generation in Fig. 5.

C. Calculation of the Cost

Having determined a finite set X of probability density functions representing the system state, and an approximation procedure for other states, we finally need the explicit evaluation of the value function. In other words, we need to compute the approximation of the actual cost values $\tilde{J}(\underline{\boldsymbol{x}}^{p,i})$ for all $\underline{\boldsymbol{x}}^{p,i} \in X$. We use a standard value iteration approach for this calculation. In the following, we refrain from the index i to provide a better readability.

First, we initialize $\tilde{J}(\underline{\boldsymbol{x}}^{p})$ for every $\underline{\boldsymbol{x}}^{p}$ by the expected position cost, ignoring the prediction assuming that a null input vector $\underline{u}^0 = \underline{0}$ generates no (or the lowest) input cost. Resulting from the separation of prediction and estimation cost, we first need to evaluate the expected cost over all possible measurements

$$\tilde{J}(\underline{\boldsymbol{x}}^{p}) \leftarrow \mathop{\mathbf{E}}_{\underline{\boldsymbol{y}}} \left\{ \tilde{J}(\underline{\boldsymbol{x}}^{e}) | \underline{\boldsymbol{x}}^{p,i} \right\} \ , \tag{12}$$

where $\underline{\boldsymbol{x}}^{e}$ is the posterior of $\underline{\boldsymbol{x}}^{p}$. Since it is not possible to evaluate this for all $\underline{\boldsymbol{y}}$, we exploit the discrete representation of $\underline{\boldsymbol{v}}_k$, and perform the calculation by a finite set of representative measurements. We do not want to maintain an explicit cost function for estimated states $\underline{\boldsymbol{x}}^{e}$ and predicted states $\underline{\boldsymbol{x}}^{p}$, but only the second. This makes it necessary to evaluate $\tilde{J}(\underline{\boldsymbol{x}}^{e})$ for every chosen representative measurement. Ignoring the prediction, thus only evaluating the position cost, the calculation of the inner cost term is given by

$$\tilde{J}(\underline{\boldsymbol{x}}^{e}) = \mathop{\mathbf{E}}_{\underline{\boldsymbol{x}}^{p}} \left\{ g(\underline{\boldsymbol{x}}^{e}, \underline{u}^0) | \underline{y}, \underline{u}^0, \underline{\boldsymbol{x}}^{p} \right\} \ ,$$

where $\underline{y}$ is one representative measurement.

After the initialization phase we iteratively update $J(\cdot)$ for every $\underline{\boldsymbol{x}}^{p}$. Equation (12) is therefore repeatedly calculated, where the inner cost term is now evaluated completely by

$$\tilde{J}(\underline{\boldsymbol{x}}^{e}) = \min_{\underline{u}} \mathop{\mathbf{E}}_{\underline{\boldsymbol{x}}^{e}} \left\{ g(\underline{\boldsymbol{x}}^{e}, \underline{u}) + \gamma \tilde{J}(\underline{\boldsymbol{x}}^{p'}) | \underline{\boldsymbol{x}}^{e}, \underline{u} \right\} \ , \tag{13}$$

where $\tilde{J}(\underline{\boldsymbol{x}}^{p'})$ is the cost function evaluated by the approximation procedure given by equation (10) using the current state of the cost function $\tilde{J}(\cdot)$. Repetitive calculation leads to an approximation of the optimal value function.

For large state spaces, it is convenient to use a finely discretized set of inputs. Since the states do not change in the cost calculation procedure, the k-nearest neighbor pairs are also fixed for a given system input. Hence, the distances have to be computed only once and the optimization (13) can be calculated very effectively.

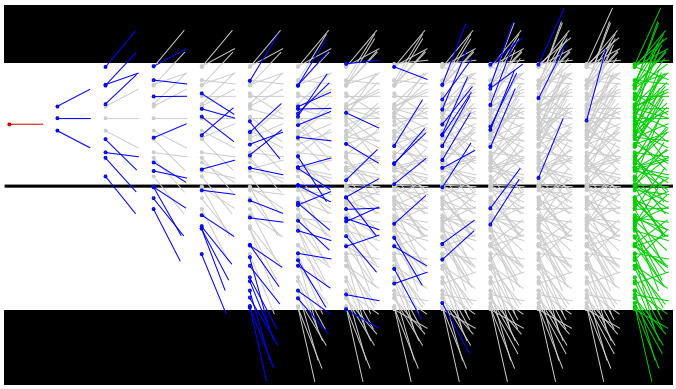


Fig. 5. Illustration of the scenario generation approach by means of an exemplary evaluation of the example system. In favor of the visualization we have only displayed the means of each probability density, with the start position (red), active branches (blue), densities added to X (gray), and finally the resulting X (green).

VI. Evaluation

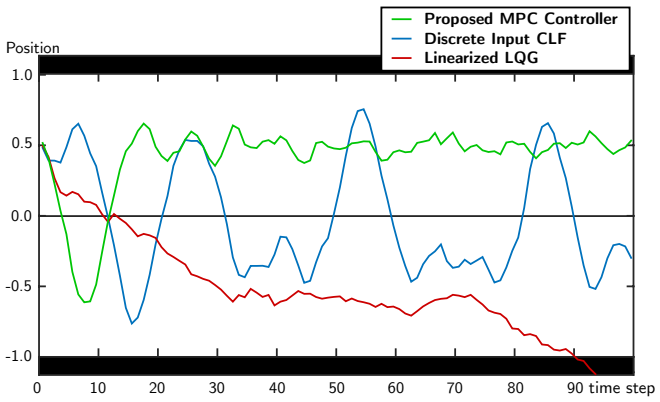
In order to illustrate the effectiveness of the proposed MPC controller based on deterministic scenario generation, we have implemented the example system introduced in Sec. II. The considered system can apply inputs $u_k \in [-0.2, 0.2]$, controlling the rotation of the vehicle. The start position is given by $\underline{x}_0 = [0.5, 0]^T$. The state estimation is initialized by $\underline{\boldsymbol{x}}_0 \sim \mathcal{N}(\underline{\hat{x}}_0, \mathbf{C}_0)$, where

$$\underline{\hat{x}}_0 = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \ , \ \mathbf{C}_0 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.03 \end{bmatrix} \ .$$

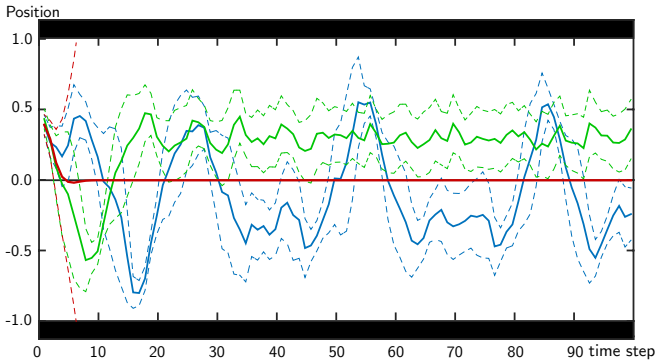
The system noise $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$ and $\underline{\boldsymbol{v}}_k \sim \mathcal{N}(0, \mathbf{C}_k^v)$ are Gaussian, zero mean, and characterized by covariance matrices

$$\mathbf{C}_k^w = \begin{bmatrix} 0.03 & 0 \\ 0 & 0.001 \end{bmatrix} \ , \ \mathbf{C}_k^v = [0.05] \ .$$

State estimation is performed by the UKF [17]. The weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ of the control cost function are both given



(a) Actual System Behavior



(b) State Estimate

Fig. 6. One exemplary evaluation of the example system for the proposed approach (green), a CLF approach for discrete control inputs (blue), and an approach based on linearization (green). (a) shows the behavior of the actual system over 100 time steps. (b) shows the state estimation, where the solid line is the mean and the dashed line indicates the uncertainty showing the $\sigma = 3$ bound.

by the identity matrix. The tunnel width, which constrains the state space, is given by two units, i.e., $x_{wall} = 1$.

We have implemented the deterministic scenario generation with the distance threshold $\theta = 0.01$ and the finite set of control inputs $\mathcal{U}_d = \{-0.2, -0.1, 0, 0.1, 0.2\}$. The system noise $\underline{\boldsymbol{w}}_k$ is discretized by $L = 5$ Diracs and the measurement noise $\underline{\boldsymbol{v}}_k$ is discretized using $L = 3$ Diracs. The step-discounting factor is chosen by $\gamma = 0.9$. Since we have not focused on chance constrains in this work, we treated them by allowing one sample to be outside the state space in the scenario generation, though evaluation of the cost yields $c = \infty$. This allows for the identification of infeasable states over the control horizon, which can be discarded before the online calculation. The offline calculation yields $|X| = 217$ states, of which as little as 73 states are feasible (i.e., $\tilde{J}(\underline{\boldsymbol{x}}^p) < \infty$) after the computation of the cumulative cost function over 15 iteration steps and 2-nearest neighbors. The online controller is implemented as stated before with continuous control inputs. First, the optimum is roughly approached by discrete inputs, followed by a gradient descent method based on the Matlab build-in constrained optimization function *fmincon*.

We compare the proposed control approach to the CLF controller for stochastic nonlinear systems with imperfect state information and finite sets of control inputs introduced in [7]. Here, we use the same finite set $\mathcal{U}_d$ and use the control

TABLE I. SIMULATION RESULTS

Method	ø cost	# crashed into wall
Proposed MPC Controller	16.87	0
Discrete Input CLF	18.75	0
Linearized LQG	56.75	40

optimization horizon $N = 3$. Furthermore, we compare both CLF controller to an LQG-based approach, where the system is linearized locally around the nominal estimate $\mathbf{E}\{\underline{\boldsymbol{x}}_k^e\}$. Since this approach uses an LQG setting, the CLF-optimal policy of the linearized system is equal to the deterministic policy, i.e., the system assumes certainty equivalence (CE), which does not hold for the original nonlinear system.

In Tab. I the simulation results of 100 Monte-Carlo simulation runs are summarized. Both approaches considering future feedback information are able to control the system without crashing. Since the proposed approach considers an infinite horizon and uses continuous system inputs, the results yield a slightly better performance compared to the short control horizon $N = 3$ and the roughly discretized system inputs. The linearization approach based on LQG and CE on the other hand leads a poor performance and frequent crashes.

In order to illustrate the control behavior, we have depicted the actual system behavior of an exemplary result for each of the three applied control approaches in Fig. 6 (a) and the state estimate, which is the actual information available to the controller, for the same runs are depicted in Fig. 6 (b), respectively. In red the linearization approach assuming CE completely loses track of the control objective, while the expected value of the estimate perfectly holds the zero line, where no observability is given. In blue, the discretized approach yields a good result. Some oversteering, brought about by the rough discretization, results in changing the side from time to time, where the controller tries to avoid the center just enough to ensure good state estimation but simultaneously minimizes the distance. A very similar result, but a bit more precisely due to the continuous-valued control input set, is shown by the proposed approach in green. The

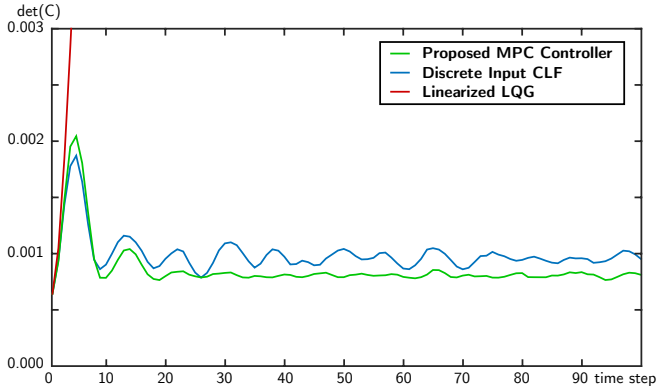


Fig. 7. Visualization of estimation quality performed during the control. The Bayesian filtering was performed for all methods by the UKF [17]. The plot shows the development of the covariance matrix over time, by means of its determinant averaged over all 100 runs. The evaluation has been performed for proposed control approach (green), a CLF approach for discrete control inputs (blue), and an approach based on linearization and the LQG controller (green).

resulting control policy is to stay in the range of 0.4 to 0.5 in order to minimize the cost, while keeping the state uncertainty low, and thus, minimizing the expected cost over the state estimate in total.

Both approaches considering the dual effect show a good coverage of estimated state and the real system behavior. This is also emphasized by Fig. 7, where we plotted the average determinant of the state estimate over 100 Monte-Carlo simulation runs. In contrast, the control policy of the CE approach is to stay at a point, where no detectability is given and thus, the uncertainty rises after few steps beyond the range of the plot.

VII. Conclusion

In this paper, a novel approach for closed-loop feedback stochastic model predictive control of time-invariant systems with continuous state space and continuous control inputs is presented. By means of an example system the challenge of controlling systems with dual effect of control is demonstrated. Thereby, it is elaborated why simultaneous estimation and control is essential to this class of control problems.

The challenge of solving the emerging stochastic dynamic programming problem is met by discretization of the cumulative cost function. A systematic scenario generation approach is introduced together with a pruning procedure, to reduce the exponential calculation load an exhaustive scenario generation would bring about. The effectiveness of the pruning, achieved by utilization of the Wasserstein distance eliminating similar states, is shown in an exemplary evaluation. The deterministic scenario generation approach guarantees the coverage of all relevant stochastic scenarios, but needs to be parametrized by the number of Diracs used in the density approximation procedure and the distance threshold that defines the minimal dissimilarity between states. Both parameters trade computational complexity against precision.

The discretized cumulative cost function enables the CLF control optimization to be calculated in terms of a one-step state prediction. We would like to emphasize that one major advantage of the presented work is that the online calculation does not need to hypothesize future measurement explicitly. These have already been considered in the offline calculation. The presented approach takes the dual effect of control into account and is able to control the introduced example system. This is demonstrated by means of a simulation.

References

[1] D. Bertsekas, "Dynamic Programming and Optimal Control", Athena Scientific, Vol. 1. 3rd ed., 1995, ISBN 1-886529-26-4.

[2] A. Brooks, A. Makarenko, S. Williams, and H. Durrant- Whyte, "Parametric POMDPs for Planning in Continuous State Spaces", *in Robotics and Autonomous Systems*, vol 54, no. 11, pp. 887–897, 2006, ISSN 0921-8890.

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

[4] G. Erdogan, L. Alexander, and R. Rajamani, "Estimation of Tire-Road Friction Coefficient Using a Novel Wireless Piezoelectric Tire Sensor", *IEEE Sensors Journal*, vol.11, no.2, pp.267–279, Feb. 2011.

[5] J. M. Maestre, P. J. van Overloop, M. Hashemy, A. Sadowska and E. F. Camacho, "Human in the Loop Model Predictive Control: An Irrigation Canal Case Study" *in Proceedings of the 53rd IEEE Conference on Decision and Control (CDC 2014)*, Los Angeles, California, USA, Dec. 2014.

[6] B. Wittenmark and C. Elevitch, "An Adaptive Control Algorithm with Dual Features", *in Preprints 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, pp. 587–592, York, UK, 1985.

[7] F. Weissel, T. Schreiter, M. F. Huber, and U. D. Hanebeck, "Stochastic Model Predictive Control of Time-Variant Nonlinear Systems with Imperfect State Information" *in Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, pp. 40-46, Aug. 2008.

[8] R. Tempo , G. Calafiore, and F. Dabbene, "Randomized Algorithms for Analysisand Control of Uncertain Systems", Springer, 2005.

[9] D. Bernardini and A. Bemporad, "Scenario-based Model Predictive Control of Stochastic Constrained Linear Systems", *in Proceedings of the 48rd IEEE Conference on Decision and Control (CDC 2014) Held Jointly with the 2009 28th Chinese Control Conference*, pp. 6333–6338, Shanghai, China, Dec. 2009.

[10] G.C. Calafiore and L. Fagiano, "Robust Model Predictive Control via Scenario Optimization", *in IEEE Transactions on Automatic Control*, vol.58, no.1, pp. 219–224, Jan. 2013.

[11] H. Heitsch and W. Römisch, "Scenario Reduction Algorithms in Stochastic Programming", *in Computational Optimization and Applications*, vol.24, pp. 187–206, 2003.

[12] G. C. Goodwin, J. Østergaard, D. E. Quevedo and A. Feuer, "A Vector Quantization Approach to Scenario Generation for Stochastic NMPC", *in Nonlinear Model Predictive Control*, Lecture Notes in Control and Information Sciences, vol. 384, pp 235–248, Springer, 2009, ISBN 978-3-642-01093-4.

[13] N. Roy, G. J. Gordon, and S. Thrun. "Finding Approximate POMDP Solutions Through Belief Compression", *in Journal of the Artificial Intelligence Research (JAIR)*, vol. 23, pp. 1–40, 2005.

[14] J. van den Berg, S. Patil, and R. Alterovitz, "Efficient Approximate Value Iteration for Continuous Gaussian POMDPs", *in Proceedings of the AAAI Conference on Artificial Intelligence*, 2012.

[15] S. Thrun, "Monte Carlo POMDPs", *Neural Information Processing Systems (NIPS) Foundation*. Vol. 12., pp. 1064–1070, 1999.

[16] E. Zhou, M. C. Fu, and S. I. Marcus., "Solving Continuous-State POMDPs Via Density Projection", *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1101-1116, 2010.

[17] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *in Proceedings of the IEEE*, Vol. 92, No. 3, pp. 401-422, Mar. 2004.

[18] J. Steinbring and U. D. Hanebeck, "S2KF: The Smart Sampling Kalman Filter", *in Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Jul. 2013.

[19] U. D. Hanebeck, "PGF 42: Progressive Gaussian Filtering with a Twist" *in Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[20] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *in IEEE Transactions on Signal Processing*, Vol. 50, No. 2, February, 2002.

[21] C. Villani, "The Wasserstein Distances", *in Optimal Transport: Old and New*, Chap. 6, pp. 93–11, Springer, 2008, ISBN 978-3-540-71049-3.

[22] H. W. Kuhn, "The Hungarian Method for the Assignment Problem", *Naval Research Logistics Quarterly*, pp. 83–97, 1955.

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

[24] 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.

[25] I. Gilitschenski and U. D. Hanebeck, "Efficient Deterministic Dirac Mixture Approximation", *in Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D. C., USA, Jun. 2013.

[26] I. Dinur and S. Safra, "On the Hardness of Approximating Minimum Vertex Cover". *in Annals of Mathematics* vol. 162, no. 1, pp. 439-485, 2005.