A Nonlinear Model Predictive Control Framework Approximating Noise Corrupted Systems with Hybrid Transition Densities

Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck

Abstract— In this paper, a framework for Nonlinear Model Predictive Control (NMPC) for heavily noise-affected systems is presented. Within this framework, the noise influence, which originates from uncertainties during model identification or measurement, is explicitly considered. This leads to a significant increase in the control quality. One part of the proposed framework is the efficient state prediction, which is necessary for NMPC. It is based on transition density approximation by hybrid transition densities, which allows efficient closed-form state prediction of time-variant nonlinear systems with continuous state spaces in discrete time. Another part of the framework is a versatile value function representation using Gaussian mixtures, Dirac mixtures, and even a combination of both. Based on these methods, an efficient closed-form algorithm for calculating an approximate value function of the NMPC optimal control problem employing dynamic programming is presented. Thus, also very long optimization horizons can be used and furthermore it is possible to calculate the value function off-line, which reduces the on-line computational burden significantly. The capabilities of the framework and especially the benefits that can be gained by incorporating the noise in the controller are illustrated by the example of a miniature walking robot following a given path.

I. INTRODUCTION

Model Predictive Control (MPC), which is sometimes also referred to as Receding or Rolling Horizon Control, has become more and more important as here not only the current system state, but also a model-based prediction of future system states is considered in the control law. In MPC, an open-loop optimal control problem for a finite N stage prediction horizon is solved according to a corresponding reward or value function. The resulting optimal control input is then applied as closed-loop control to the system.

As the well understood and widely used Model Predictive Control for linear system models [14] together with linear or quadratic reward functions is not always sufficient to meet the steadily growing requirements on the control quality, nonlinear system models and reward functions need to be considered in the control, which is then called Nonlinear Model Predictive Control (NMPC). The increased computational demand due to the consideration of nonlinearities has been mitigated in the last years by advances in the employed algorithms to solve the necessary open-loop optimization [13] as well as the steadily increasing available computation power for control processes [5]. Even if the incorporation of nonlinearities leads to a significant increase in the control quality, most approaches do not consider

Florian Weissel, Marco F. Huber, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory, Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany. {weissel|marco.huber|uwe.hanebeck}@ieee.org
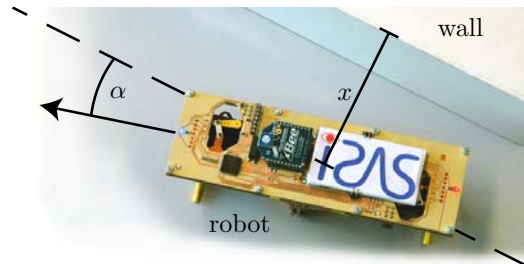
Fig. 1. Miniature walking robot [17].

the influence of noise on the system [3] especially in the important continuous state space case. In [9], a solution for linear system is given, that considers uncertainties. In [4], the stochastic system behavior is considered in the control, but the control is only determined in the vicinity of the deterministic solution, which just leads to a locally optimal solution. In [12], an approach to infinite-horizon optimal control is presented, where a radial-basis-function network is used to discretize the continuous state space. In this approach, the noise influence is considered, but it suffers, as any discretization, from the curse of dimensionality. Additionally, it is computational very demanding due to a required high-dimensional matrix inversion.

Many technical systems are only able to handle a finite set of control inputs even if they have a continuous-valued state space. This may, for example be a robot as depicted in Fig. 1, whose state space comprises its posture, but which is still just able to handle commands like turn left / right or move straight. Another example would be a sensor network, whose state space comprises continuous-valued quantities to be measured with related positions, but whose inputs are just certain discrete measurement requests. Additionally, the models and sensor readings in most of these systems are often to a high degree uncertain. Therefore, in this paper a framework for discrete-time Nonlinear Model Predictive Control for continuous state spaces and a finite set of control inputs is presented that explicitly incorporates modeling and measurement uncertainties. For calculating the necessary predicted system states, a transition density approximation technique that describes the system transition densities based on hybrid Dirac and Gaussian densities as proposed in [7] is employed. This approximate system description allows closed-form calculation of the predicted system states with constant complexity. As an additional part of this framework, an extremely flexible representation of the reward function based on Dirac mixtures, Gaussian mixtures as well as a combination of both is presented. This representation is very

expressive due to the universal approximation property [11] of Gaussian mixtures that is even enhanced by the Dirac mixtures. To decrease the on-line computational burden significantly, a value function approximation scheme employing dynamic programming is introduced, that seamlessly fits in the proposed framework. By combining these techniques, an efficient integrated closed-form approach to NMPC for noise-affected systems with novel abilities is gained.

The remainder of this paper is structured as follows: In the next section, the considered NMPC problem is described together with an example from the field of mobile robot control. In Section III, the efficient closed-form prediction approach for nonlinear systems based on transition density approximation is derived. Techniques modeling the reward function are described in Section IV. Then, in Section V, a value function approximation scheme based on dynamic programming is introduced. In Section VI, three different kinds of NMPC controllers are compared based on simulations employing the example system, which has been introduced in previous sections. The paper closes with conclusions and perspectives on future work.

II. PROBLEM FORMULATION

In this paper, discrete-time systems of the form

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

are considered, where $\underline{x}_k$ and $\underline{u}_k$ denote the random vectors corresponding to the system state and the applied control input. $\underline{a}_k(\cdot)$ is a nonlinear, possibly time-variant function, where $\underline{w}_k$ denotes the additive white noise.

Example System
A mobile miniature walking robot (Fig. 1) is supposed to move along a given trajectory, e.g. along a wall, with constant velocity. This robot is able to superimpose left and right turns onto the forward motion. The robot's motion can be modeled similar to the motion of a two-wheeled differential-drive robot, which leads to the nonlinear discrete-time system equation

$$\begin{aligned} x_{k+1} &= x_k + v \cdot T \cdot \sin(\alpha_k) + w_k^x \ , \\ \alpha_{k+1} &= \alpha_k + u_k + w_k^\alpha \ , \end{aligned} \qquad (2)$$

where $\underline{x}_k = [x_k, \alpha_k]^{\mathsf{T}}$, with the distance to the wall x_k and its orientation relative to the wall α_k. The robot's constant velocity is v, T is the sampling interval, and w_k^x, w_k^α denote the noise influence on the system. The input u_k is the steering action, i.e., turn left, turn right, or go straight. Furthermore, the robot is equipped with sensors for measuring the distance y_k^x and orientation y_k^α with respect to the wall according to

$$\begin{aligned} y_k^x &= x_k + v_k^x \ , \\ y_k^\alpha &= \alpha_k + v_k^\alpha \ , \end{aligned} \qquad (3)$$

where v_k^x and v_k^α are the measurement noise.

At any time step k, an open-loop optimal control problem is solved for an N step prediction horizon and a corresponding cumulative value function

$$J_k(\underline{x}_k) =$$
$$\max_{\underline{u}_{k,0:N-1}} \mathrm{E}_{\underline{x}_{k,1:N}} \left\{ g_N(\underline{x}_{k,N}) + \sum_{n=0}^{N-1} g_n(\underline{x}_{k,n}, \underline{u}_{k,n}) \right\} \ , \qquad (4)$$

with $\underline{x}_k = \underline{x}_{k,0}$. $J_k(\underline{x}_k)$ comprises the step reward $g_n(\underline{x}_{k,n}, \underline{u}_{k,n})$ depending on the predicted system states $\underline{x}_{k,n}$ and the corresponding control inputs $\underline{u}_{k,n}$, as well as a terminal reward $g_N(\underline{x}_{k,N})$. The optimal control input $\underline{u}_k^*$, which is determined according to

$$\underline{u}_k^*(\underline{x}_k) = \arg\max_{\underline{u}_k} V_k(\underline{x}_k, \underline{u}_k) \ , \qquad (5)$$

with $\underline{u}_k = \underline{u}_{k,0}$ and

$$V_k(\underline{x}_k, \underline{u}_k) =$$
$$\max_{\underline{u}_{k,1:N-1}} \mathrm{E}_{\underline{x}_{k,1:N}} \left\{ g_N(\underline{x}_{k,N}) + \sum_{n=0}^{N-1} g_n(\underline{x}_{k,n}, \underline{u}_{k,n}) \right\} \ , \qquad (6)$$

is then applied to the system. In the next time step $k+1$, the whole procedure is repeated.

For most nonlinear systems, the analytic evaluation of (6) is not possible. This is due to the required prediction of the system state for a noise-affected nonlinear system as well as the necessity of calculating expected values. Both operations typically cannot be performed in closed form. In the next sections, an integrated approach is presented that overcomes these two problems and allows closed-form calculations.

III. STATE PREDICTION WITH HYBRID DENSITIES

As described above, in stochastic NMPC the system state $\underline{x}_{k,1:N}$ needs to be predicted over an N step horizon for all possible inputs $\underline{u}_{k,0:N-1}$. For a certain k, the probability density $\tilde{f}_{n+1}^x(\underline{x}_{n+1})$ of the system state $\underline{x}_{n+1}$ for the next time step $n+1$, and thus recursively for the next N time steps, can be computed utilizing the so-called Chapman-Kolmogorov equation [1]

$$\tilde{f}_{n+1}^x(\underline{x}_{n+1}) = \int_{\mathbb{R}^d} \tilde{f}_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n) \, \tilde{f}_n^x(\underline{x}_n) \, \mathrm{d}\underline{x}_n \ , \qquad (7)$$

with the input dependent transition density $\tilde{f}_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n) = f_n^w(\underline{x}_{n+1} - \underline{a}_n(\underline{x}_n, \underline{u}_n))$, where $f_n^w(\cdot)$ denotes the probability density function of the system noise $\underline{w}_n$ and $\underline{a}_n(\cdot)$ the nonlinear system function from (1). For linear systems with Gaussian random variables, the Kalman filter [8] provides an exact solution to (7), since this equation is reduced to the evaluation of an integral over a product of two Gaussian densities, which is analytically solvable.

As an exact closed-form solution to (7) is typically not available for nonlinear systems and numerical solutions are highly demanding, an approximate calculation is inevitable. In the following, true densities are indicated with a tilde, the approximations are not. One common approach in the context of NMPC is linearizing the system and then applying the Kalman filter [10]. The resulting single Gaussian density is typically not sufficient for approximating $\tilde{f}_{n+1}^x(\underline{x}_{n+1})$. Hence, we propose to approximate the transition density $\tilde{f}_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n)$ in (7) by a *hybrid density* [7] of the form

$$f_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n) = \sum_{i=1}^L \omega_i \cdot \delta(\underline{x}_n - \underline{\mu}_i^{(n)}) \cdot f(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \underline{\nu}_i),$$
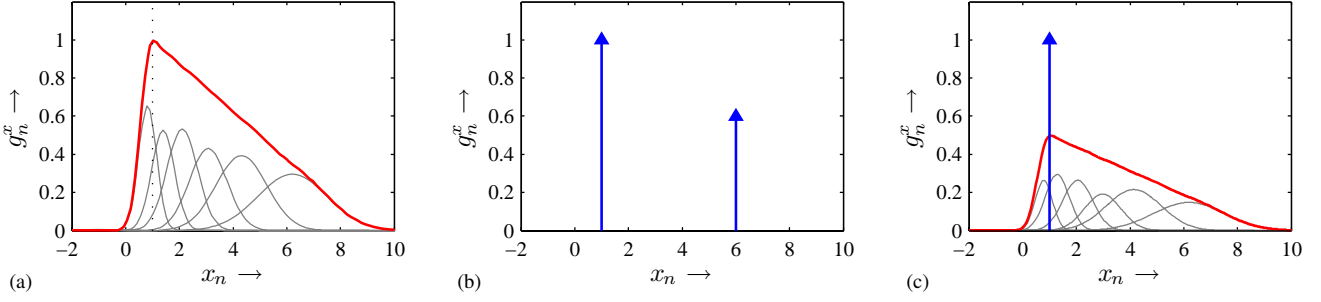$$\qquad (8)$$

Fig. 2. Asymmetric reward functions: (a) Gaussian mixture reward function, (b) Dirac mixture reward function (c) combination of Gaussian mixture and Dirac mixture reward function.

where L is the number of components, ω_i are weighting coefficients with $\omega_i > 0$, $\underline{\mu}_i^{(n)}$ and $\underline{\mu}_i^{(n+1)}$ are location parameters, and $\underline{\nu}_i$ are shape parameters of $f(\underline{x}_{n+1} - \underline{\mu}_i^{(n)}; \underline{\nu}_i)$. This hybrid density consists of two different types of analytic densities: Dirac delta distributions $\delta(\underline{x}_n - \underline{\mu}_i^{(n)})$ and another density $f(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \underline{\nu}_i)$ that matches the probability density function of the system noise. In the following, Gaussian noise and thus Gaussian densities $f(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \underline{\nu}_i) = \mathcal{N}(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \mathbf{C}_i^{(n+1)})$, with mean $\underline{\mu}_i^{(n+1)}$ and covariance matrix $\mathbf{C}_i^{(n+1)}$ are considered. Methods to choose the optimal parameters of (8) in an efficient fashion are described in [7]. By approximating the transition density $\tilde{f}_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n)$ of the system by a hybrid density as described above, the evaluation of (7) can be carried out very easily in closed form for arbitrary well-defined probability density functions $\tilde{f}_n^x(\underline{x}_n)$. The resulting probability density function $f_{n+1}^x(\underline{x}_{n+1})$ is a Gaussian mixture of the form

$$f_{n+1}^x(\underline{x}_{n+1}) = \sum_{i=1}^{L} \omega_i^{(n+1)} \cdot \mathcal{N}\left(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \mathbf{C}_i^{(n+1)}\right) ,$$

where the weights $\omega_i^{(n+1)}$ can be calculated by evaluating $\tilde{f}_n^x(\underline{x}_n)$ at the $\underline{\mu}_i^{(n)}$ of the transition density. The parameters $\underline{\mu}_i^{(n+1)}$ and $\mathbf{C}_i^{(n+1)}$ of $f_{n+1}^x(\underline{x}_{n+1})$ correspond to the parameters of the Gaussians of the transition density.

In contrast to our approach in [16], where pure Gaussian mixture transition densities are used and thus demanding off-line optimizations are necessary, which limit it to time-invariant systems, here also time-variant systems with non-stationary noise can be handled easily.

IV. REWARD FUNCTION REPRESENTATIONS

Besides being able to predict the system state, it is also important to be able to describe the reward functions as introduced in (4) in a versatile way. In the following, reward functions $g_n(\underline{x}_n, \underline{u}_n)$ are considered that are additively decomposable into a state-dependent and an input-dependent part according to

$$g_n(\underline{x}_n, \underline{u}_n) = g_n^x(\underline{x}_n) + g_n^u(\underline{u}_n) .$$

For modeling the state-dependent part $g_n^x(\underline{x}_n)$, we propose a reward function representation by means of Gaussian mixtures similar to the one described in [16]. This representation is already very expressive due to the universal

approximation property of Gaussian mixtures [11]. If not only soft objectives are to be modeled, which can be done very easily with Gaussian mixtures, but dedicated system states are to be described, this can be done very well with Dirac mixtures. Obviously, also a combination of the Gaussian mixture and Dirac mixture representation can be used and both the Gaussian mixtures and Dirac mixtures may have arbitrary parameters, e.g. negative weights ω.

Example System: Reward Function

A reward function as depicted in Fig. 2 (a) can be used to let the robot move at a certain distance to the wall (e.g. $\check{x}_n = 1$), where being closer to the wall ($x_n^{Wall} = 0$) is considered much less desirable than being farther away.

If the robot were to move at certain well-defined distances to the wall (e.g. $\check{x}_n^a = 1$ and $\check{x}_n^b = 6$) a reward function as depicted in Fig. 2 (b) could be used.

Obviously, also a combination of the Gaussian mixture and the Dirac mixture representation can be used (Fig. 2 (c)).

If not only the distance to the wall, but also the orientation is to be incorporated into the reward function, this can be done by extending the dimensionality of the reward function, which leads to a reward function as depicted in Fig. 3. Here, the robot is also driven to move in parallel to the wall.

For a Dirac mixture reward function $g_n^x(\underline{x}_n)$, the expected value $\mathrm{E}_{\underline{x}_n}\{g_n^x(\underline{x}_n)\}$, which is necessary to compute (6), can be calculated in closed form according to

$$\mathrm{E}_{\underline{x}_n}\{g_n^x(\underline{x}_n)\}$$
$$= \int_{\mathbb{R}^d} f_n^x(\underline{x}_n) \cdot g_n^x(\underline{x}_n)\, \mathrm{d}\underline{x}_n$$
$$= \int_{\mathbb{R}^d} \sum_{i=1}^{L} \omega_i^{(n)} \mathcal{N}(\underline{x}_n - \underline{\mu}_i^{(n)}; \mathbf{C}_i^{(n)})$$
$$\cdot \sum_{j=1}^{M} \omega_j^{(n)}\, \delta(\underline{x}_n - \underline{\mu}_j^{(n)})\, \mathrm{d}\underline{x}_n$$
$$= \sum_{i=1}^{L}\sum_{j=1}^{M} \omega_i^{(n)}\omega_j^{(n)} \mathcal{N}(\underline{\mu}_j^{(n)} - \underline{\mu}_i^{(n)}; \mathbf{C}_i^{(n)}) , \quad (9)$$

where $f_n^x(\underline{x}_n)$ denotes the Gaussian mixture probability density function of the system state. The closed-form calculation in case of Gaussian mixture reward functions is shown in [16]. In case of mixed reward functions, the solution can be gained by calculating the Dirac mixture part and the Gaussian mixture part separately.
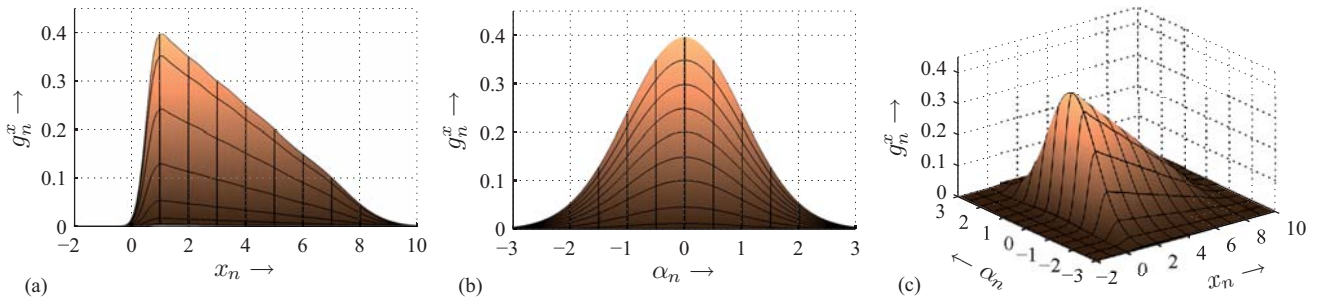
Fig. 3. Asymmetric two-dimensional reward function consisting of six components with a maximum at $\breve{x}_n = 1$ and $\breve{\alpha}_n = 0$.

The input dependent part of the reward function $g_n^u(\underline{u}_n)$ can be modeled with any kind of function or a lookup-table since there is just a finite number of discrete $\underline{u}_n$.

By using the efficient state prediction presented in Section III together with the reward function representations presented above, the calculation of the value function and the optimal control as described in (5) and (6) can be carried out, as shown, in closed form.

V. VALUE FUNCTION APPROXIMATION USING DYNAMIC PROGRAMMING

In case of short optimization horizons, the techniques described above allow an efficient calculation of the open-loop optimal control problem within NMPC, where the straightforward solution is an exhaustive search of a tree with depth N. For longer optimization horizons, the straightforward calculation becomes unfeasible as the computational demand increases exponentially with the length N of the optimization horizon. This can be resolved by calculating the value function (4) in an efficient approximate fashion using dynamic programming (DP), which is possible since the value function only consists of additive terms [2]. Employing DP, (4) can be calculated in a backward fashion according to

$$J_N(\underline{x}_N) = g_N(\underline{x}_N) \ ,$$
$$J_n(\underline{x}_n) = \max_{\underline{u}_n} \{V_n(\underline{x}_n, \underline{u}_n)\} \ , \ n = N-1, \ldots, 0 \ , \tag{10}$$

with

$$V_n(\underline{x}_n, \underline{u}_n) = g_n^x(\underline{x}_n) + g_n^u(\underline{u}_n) + \mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$$

for a fixed k. Using this recursion, the computational demand just increases linearly with N. Additionally, this allows off-line calculation of (6), thus only (5) needs to be evaluated on-line, which increases the on-line performance significantly.

For calculating $J_n(\underline{x}_n)$, two main operations have to be executed per step:

A. First, $\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$ has to be calculated.
B. Second, $V_n(\underline{x}_n, \underline{u}_n)$ has to be maximized with respect to $\underline{u}_n$.

A. Calculation of $\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$

Employing the fact that system (1) is represented by a hybrid transition density $f_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n)$ as introduced in Section III, an approximation of the conditional expected

value $\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$ that depends on the state $\underline{x}_n$ and the input $\underline{u}_n$ can be calculated according to

$$\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$$
$$\approx \int_{\mathbb{R}^d} f_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n) \cdot J_{n+1}(\underline{x}_{n+1}) \, \mathrm{d}\underline{x}_{n+1}$$
$$= \int_{\mathbb{R}^d} \sum_{i=1}^{L} \omega_i \cdot \delta(\underline{x}_n - \underline{\mu}_i^{(n)}) \cdot \mathcal{N}(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \mathbf{C}_i^{(n+1)})$$
$$\cdot J_{n+1}(\underline{x}_{n+1}) \, \mathrm{d}\underline{x}_{n+1}$$
$$= \sum_{i=1}^{L} \omega_i \cdot \delta(\underline{x}_n - \underline{\mu}_i^{(n)})$$
$$\cdot \underbrace{\int_{\mathbb{R}^d} \mathcal{N}(\underline{x}_{n+1} - \underline{\mu}_i^{(n+1)}; \mathbf{C}_i^{(n+1)}) \cdot J_{n+1}(\underline{x}_{n+1}) \, \mathrm{d}\underline{x}_{n+1}}_{=:z_i}$$
$$= \sum_{i=1}^{L} \omega_i z_i \cdot \delta(\underline{x}_n - \underline{\mu}_i^{(n)}) =: K_n(\underline{x}_n, \underline{u}_n) \ , \tag{11}$$

where the result of this calculation is a Dirac mixture. Additionally, if $J_{n+1}(\underline{x}_{n+1})$ is approximated by a Dirac mixture (DM), a Gaussian mixture (GM), or a sum of both (which will be shown in the next section), the calculation of (11) can be performed very easily and in closed form as it is just the multiplication of two Gaussian mixtures or a Gaussian mixture and a Dirac mixture with a subsequent marginalization[1].

B. Maximization

In the following, the necessary maximization of

$$V_n(\underline{x}_n, \underline{u}_n) \approx \underbrace{g_n^x(\underline{x}_n)}_{\text{GM / DM}} + \underbrace{g_n^u(\underline{u}_n)}_{\text{DM}} + \underbrace{K_n(\underline{x}_n, \underline{u}_n)}_{\text{DM}}$$

with respect to $\underline{u}_n$ is described.

$g_n^x(\underline{x}_n)$ has no impact on the maximization as it does not depend on $\underline{u}_n$. $K_n(\underline{x}_n, \underline{u}_n)$ is, as described above, a Dirac mixture depending on continuous-valued $\underline{x}_n$ and discrete $\underline{u}_n$. Thus, a maximization of this part with respect to $\underline{u}_n$ is easily possible, as long as the $\underline{\mu}_i^{(n)}$ of the Diracs are identical for all $\underline{u}_n$. This restriction can be easily met if it is considered during the approximation of $f_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n)$. In this case, the maximization can be carried out individually for each set of Diracs with equal $\underline{\mu}_i^{(n)}$.

[1]The calculation for a Gaussian mixture and a Dirac mixture is equivalent to (9), the calculation for two Gaussian mixtures is described in [16].
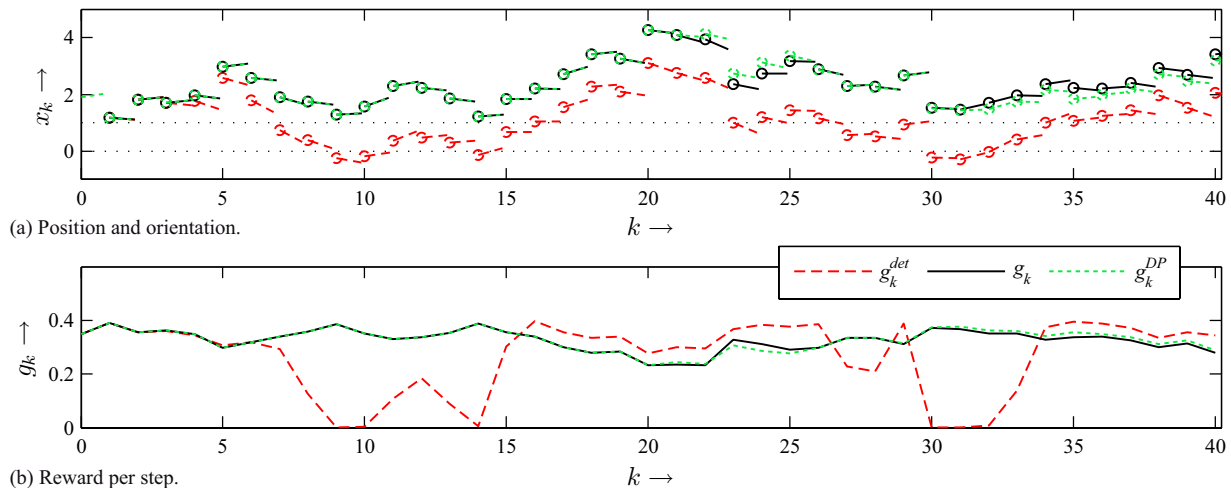
Fig. 4. 40 step simulation. (red dashed line: deterministic NMPC, black solid line: stochastic NMPC, green dotted line: stochastic NMPC with DP).

At first sight, the maximization of the sum of $K_n(\underline{x}_n, \underline{u}_n)$ and $g_n^u(\underline{u}_n)$ poses a problem in case $g_n^u(\underline{u}_n)$ is an arbitrary function depending on $\underline{u}_n$, but as $g_n^u(\underline{u}_n)$ does not depend on $\underline{x}_n$ it is thus constant with respect to $\underline{x}_n$ and can be easily approximated by a Dirac mixture. If this Dirac mixture is chosen in such a way that its components are at the very same position as the components of $K_n(\underline{x}_n, \underline{u}_n)$, again the maximization can be carried out easily.

If $f_{\underline{u}_n}^T(\underline{x}_{n+1}|\underline{x}_n)$ and $g_n^u(\underline{u}_n)$ are approximated as described above, $\max_{\underline{u}_n}\{J_n(\underline{x}_n, \underline{u}_n)\}$ from (10) can be approximated by the sum of a Dirac mixture, arising from $g_n^u(\underline{u}_n)$, $\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$ and the Dirac mixture part of $g_n^x(\underline{x}_n)$, and a Gaussian mixture, arising from the Gaussian mixture part of $g_n^x(\underline{x}_n)$, and is then of the same structure as the reward function depicted in Fig. 2(c). Thus, an approximation of $\mathrm{E}_{\underline{x}_{n+1}}\{J_{n+1}(\underline{x}_{n+1})|\underline{x}_n\}$ can be calculated, as shown, efficiently in closed form.

VI. SIMULATIONS

In order to illustrate the modeling capabilities of the introduced NMPC framework as well as the benefits gained by directly considering the noise in the control, several simulations have been conducted using the example of a miniature robot walking along a wall. The considered non-linear system is given by (2) and (3), with $v \cdot T = 1$ and inputs $u_k \in \{-0.2, 0, 0.2\}$ (turn left/right, go straight). The noise influence on the system $\boldsymbol{w}_k^x$ and $\boldsymbol{w}_k^\alpha$ is zero-mean white Gaussian noise with standard deviation $\sigma_w^x = 0.5$ and $\sigma_w^\alpha = 0.05 \approx 2.9°$. The measurement noise is also zero-mean white Gaussian noise with standard deviation $\sigma_v^x = 0.5$ and $\sigma_v^\alpha = 0.05$. All simulations are performed for a $N = 4$ step prediction horizon, with a reward function according to (4), where $g_N(\boldsymbol{x}_N, \boldsymbol{\alpha}_N)$ is the function depicted in Fig. 3 and $g_n(\boldsymbol{x}_n, \boldsymbol{\alpha}_n) = g_N(\boldsymbol{x}_N, \boldsymbol{\alpha}_N), \forall n$.

To evaluate the benefits of the proposed NMPC framework, three different kinds of simulation are conducted:

1) Calculation of the control input without any consideration of noise (deterministic NMPC): The deterministic

control neglecting the noise influence has been calculated as a benchmark.

2) Straightforward calculation of the control input with consideration of the noise influence (stochastic NMPC): The optimal control input with explicit consideration of noise has been calculated employing the techniques presented in Section III and Section IV. Therefore, all calculations could be executed analytically without the need for any numerical methods. The employed approximate transition densities with 25 elements have been calculated on-line. Still, this approach has the drawback that the computational demand for the optimal control problem increases exponentially with the length of the horizon N. Thus, this approach is only suitable for short horizons.

3) Calculation of the control input with the proposed value function approximation scheme employing dynamic programming (stochastic NMPC with Dynamic Programming): The control input is calculated using the value function approximation technique presented in Section V. Thus, the noise influence is also considered directly in the control and additionally longer prediction horizons could be easily used without any impact on the on-line computation performance. The resulting approximate value function consists of 200 Dirac components from $K_{k,0}(\cdot)$ and 6 Gaussian components from $g_{k,0}^x(\cdot)$.

In Fig. 4 (a), the first 40 steps of an example simulation run are shown. The distance to the wall x_k is depicted by the position of the circles, the orientation α_k by the orientation of the arrows. For each simulation run, a particular noise realization has been used that was applied to all the different controllers. It can be clearly seen that the robot under deterministic control behaves very differently from the other two. The deterministic controller just tries to move the robot to the maximum of the reward function at $\check{x}_k = 1$ and $\check{\alpha}_k = 0$, while it totally neglects the asymmetry of the reward function. The two stochastic controllers both lead to a larger distance from the wall, as they consider the noise influence in conjunction with the non-symmetric reward function. Both stochastic controllers behave similarly, which illustrates the

TABLE I

SIMULATION RESULTS

	det. control	stoch. control	stoch. DP control
average reward	0.2528	0.2922	0.2927
normalized av. reward	100%	115.6%	115.8%

good performance of the value function approximation techniques proposed in Section V. In Fig. 4 (b), the evaluation of the reward function for each step is shown. As expected, both stochastic controllers perform much better, i.e., they generate a higher average reward than the deterministic one. This finding has been validated by a series of 100 Monte Carlo simulations with different noise realizations and initial values. The uniformly distributed initial values are sampled from the intervals $x_0 \in [0, 6]$ and $\alpha_0 \in [-\pi/4, \pi/4]$. In Table I, the average step rewards of the 100 simulations are shown with 40 steps each. To facilitate the comparison also normalized average step rewards are given. It can be clearly seen that the stochastic controllers outperform the deterministic one by over 15% in terms of reward. In more than 86% of the runs, the stochastic controllers gave better results than the deterministic one. Therefore, the performance increase by the stochastic controllers is highly significant with a p-Value of below 10^{-13}.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, a novel framework for Nonlinear Model Predictive Control is introduced that explicitly considers the noise influence on the system. This framework uses a stochastic state prediction method based on transition density approximation by hybrid densities, which allows closed-form computation and, if needed, fast on-line approximation and thus enables the framework to handle time-variant system function. Another important aspect of the proposed NMPC framework is the modeling of the reward function by Gaussian mixtures, Dirac mixtures, and even a combination of both, which leads to a level of flexibility far beyond the traditional representations. By employing matching representations for both the approximate transition densities and the reward functions, a closed-form solution is obtained. Based on the transition density approximation and the reward function representation as well as employing dynamic programming, a closed-form algorithm for calculating the value function of the related optimal control problem is presented that permits its off-line calculation. This significantly reduces the required on-line computational demand and also allows the use of long prediction horizons. The benefits gained by the consideration of noise in the controller as well as the effectiveness of the presented framework and its algorithms are illustrated with simulations of a miniature robot walking along a wall.

One interesting future extension will be the incorporation of the state estimation in the control, which is important for nonlinear systems, as here the separation principle does not hold. An additional interesting aspect will be the consideration of effects of inhomogeneous noise, i.e., noise with state

and/or input dependent noise levels. Here, the consideration of the stochastic behavior of the system is expected to have an even greater impact on the control quality. Also the extension to new application fields is intended. Of special interest is the extension to the related emerging field of Model Predictive Sensor Scheduling [6], [15], which is of special importance, e.g. in sensor-actuator-networks.

REFERENCES

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

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

[3] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Springer-Verlag London Ltd., Jun. 2004.

[4] M. P. Deisenroth, F. Weissel, T. Ohtsuka, and U. D. Hanebeck, "Online-Computation Approach to Optimal Control of Noise-Affected Nonlinear Systems with Continuous State and Control Spaces," in *Proceedings of the European Control Conference (ECC 2007)*, Kos, Greece, Jul. 2007.

[5] R. Findeisen and F. Allgöwer, "An Introduction to Nonlinear Model Predictive Control," in *Summerschool on "The Impact of Optimization in Control", Dutch Institute of Systems and Control (DISC)*, C. Scherer and J. Schumacher, Eds., 2002, pp. 3.1–3.45.

[6] Y. He and E. K. P. Chong, "Sensor Scheduling for Target Tracking in Sensor Networks," in *Proc. of the 43rd IEEE Conference on Decision and Control*, vol. 1, Dec. 2004, pp. 743–748.

[7] M. F. Huber and U. D. Hanebeck, "The Hybrid Density Filter for Nonlinear Estimation based on Hybrid Conditional Density Approximation," in *10th International Conference on Information Fusion (Fusion 2007)*, Quebec, Canada, Jul. 2007.

[8] R. E. Kalman, "A new Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[9] B. Kouvaritakis, M. Cannon, and V. Tsachouridis, "Recent developments in stochastic MPC and sustainable development," *Annual Reviews in Control*, vol. 28, pp. 23–35, 2004.

[10] J. H. Lee and N. L. Ricker, "Extended Kalman Filter Based Nonlinear Model Predictive Control," in *Industrial & Engineering Chemistry Research*. ACS, 1994, pp. 1530–1541.

[11] V. Maz'ya and G. Schmidt, "On approximate approximations using Gaussian kernels," *IMA Journal of Numerical Analysis*, vol. 16, no. 1, pp. 13–29, 1996.

[12] D. Nikovski and M. Brand, "Non-Linear Stochastic Control in Continuous State Spaces by Exact Integration in Bellman's Equations," in *Proc. of the 2003 International Conference on Automated Planning and Scheduling*, Jun. 2003, pp. 91–95.

[13] T. Ohtsuka, "A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2003.

[14] S. J. Qin and T. A. Badgewell, "An Overview of Industrial Model Predictive Control Technology," *Chemical Process Control*, vol. 93, pp. 232–256, 1997.

[15] A. V. Savkin, R. J. Evans, and E. Skafidas, "The Problem of Optimal Robust Sensor Scheduling," in *Proc. of the 39th IEEE Conference on Decision and Control*, vol. 4, Sydney, Australia, Dec. 2000, pp. 3791–3796.

[16] F. Weissel, M. F. Huber, and U. D. Hanebeck, "Efficient Control of Nonlinear Noise–Corrupted Systems Using a Novel Model Predictive Control Framework," in *Proceedings of the 2007 American Control Conference (ACC 2007)*, New York City, New York, Jul. 2007.

[17] ——, "Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, California, Nov. 2007.