

# Efficient Control of Nonlinear Noise-Corrupted Systems Using a Novel Model Predictive Control Framework

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

**Abstract**— Model identification and measurement acquisition is always to some degree uncertain. Therefore, a framework for Nonlinear Model Predictive Control (NMPC) is proposed that explicitly considers the noise influence on nonlinear dynamic systems with continuous state spaces and a finite set of control inputs in order to significantly increase the control quality. Integral parts of NMPC are the prediction of system states over a finite horizon as well as the problem specific modeling of reward functions. For achieving an efficient and also accurate state prediction, the introduced framework uses transition densities approximated by means of axis-aligned Gaussian mixtures. The representation power of Gaussian mixtures is also used to model versatile reward functions. Thus, together with the prediction technique a closed-form calculation of the optimization problems arising from NMPC is possible. Additionally, an efficient algorithm for calculating an approximate value function of the corresponding optimal control problem employing dynamic programming is presented. Thus, the value function can be calculated off-line, which reduces the on-line computational burden significantly and also permits the use of long optimization horizons. 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 two-wheeled differential-drive mobile robot following a given path.

## I. INTRODUCTION

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

To gain an even higher quality of control, the well understood and widely used MPC for linear system models [14], which uses linear or quadratic reward functions, is not always sufficient. By incorporating nonlinear system models and nonlinear reward functions, steadily growing requirements on the control quality can be met. The typically significant increase in computational demand arising from the nonlinearities has been mitigated in the last years by the steadily increasing available computation power for control processes [6] and advances in the employed algorithms to solve the connected open-loop optimization [13].

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

Still, most approaches, especially for the important case of continuous state space, do not explicitly consider the influence of noise on the system [3]. This obviously leads to insufficient solutions especially for highly nonlinear systems and reward functions. In [5], an extension of the deterministic reward function by a term considering the noise is presented. In [12], an approach for infinite horizon optimal control is presented, where a continuous state space is discretized by means of a radial-basis-function network. This approach leads to a consideration of the noise influence, but suffers as any discretization, from the curse of dimensionality.

In many fields of application, as robotics or sensor-actuator-networks, discrete-time controllers for systems with continuous-valued state spaces but a finite set of control inputs are of special importance. For example, the posture of a robot is typically continuous-valued but its control inputs are just discrete (e.g. turn left / right or move straight). Even if the control inputs are continuous valued in many cases they can be easily discretized in a meaningful fashion.

Basing on the fundamentals derived in [16], 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 utilizes dynamic programming and an efficient state prediction. The prediction method for nonlinear stochastic models is founded on the approximation of the involved system transition densities by axis-aligned Gaussian mixture densities [8], enabling closed-form estimation resulting in an accurate Gaussian mixture approximation of the predicted state density. To lower the computational demands for approximating multi-dimensional transition densities, the so-called modularization for complexity reduction purposes is employed. As an additional part of this framework, an extremely flexible representation of the reward function based on Gaussian mixtures is presented. This representation is very expressive due to the universal approximation property [11] of Gaussian mixtures. In order to decrease the on-line computational burden significantly, a value function approximation scheme employing dynamic programming is introduced, that fits in the proposed framework seamlessly. Combining the various techniques, an efficient integrated closed-form approach to MPC for nonlinear 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 and complexity reduction is derived. Tech-

niques for 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 the following, we consider a nonlinear discrete-time system

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

where  $\underline{\mathbf{x}}_k$  denotes the vector-valued random variable of the system state at time  $k$ ,  $\underline{\mathbf{u}}_k$  the applied control input, and  $\underline{a}(\cdot)$  a nonlinear, time-invariant function.  $\underline{\mathbf{w}}_k$  denotes the white stationary, but not necessarily Gaussian, noise affecting the system additively *element-wise*, i.e., the elements of  $\underline{\mathbf{w}}_k$  are processed in  $\underline{a}(\cdot)$  just additively. For details see Section III.

### Example System

For illustration purpose, a special realization of (1) is considered throughout the paper. A two-wheeled differential-drive robot that is supposed to drive along a given trajectory, e.g. along a wall, can be modeled by the discrete-time system equation

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + v \cdot T \cdot \sin(\alpha_k) + \mathbf{w}_k^x, \\ \alpha_{k+1} &= \alpha_k + u_k, \end{aligned} \quad (2)$$

where the system state  $\mathbf{x}_k$  comprises the robot's distance to the wall  $x_k$  and its orientation relative to the wall  $\alpha_k$ .  $v$  is the robot's constant velocity,  $T$  the sampling interval, and  $\mathbf{w}_k^x$  denotes the noise influence on the system. The input  $u_k$  is a steering action, i.e., a change of direction of the robot. Furthermore, the robot is equipped with sensors to measure distance  $y_k^x$  and orientation  $y_k^\alpha$  with respect to the wall according to

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

where  $v_k^x$  and  $v_k^\alpha$  describe the measurement noise.

For every control step of NMPC, i.e., typically for every time step  $k$ , an open-loop optimal control problem for a finite  $N$  step prediction horizon is solved, where the optimality is defined by a cumulative value function  $J_k(\underline{\mathbf{x}}_k)$

$$J_k(\underline{\mathbf{x}}_k) = \max_{\underline{\mathbf{u}}_k} V_k(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k) = \max_{\substack{\underline{\mathbf{u}}_k, \dots, \\ \underline{\mathbf{u}}_{k+N-1}}} \mathbb{E}_{\substack{\underline{\mathbf{x}}_{k+1}, \dots, \\ \underline{\mathbf{x}}_{k+N}}} \left\{ g_N(\underline{\mathbf{x}}_{k+N}) + \sum_{n=k}^{k+N-1} g_n(\underline{\mathbf{x}}_n, \underline{\mathbf{u}}_n) \right\}, \quad (4)$$

comprising the step reward  $g_n(\underline{\mathbf{x}}_n, \underline{\mathbf{u}}_n)$  depending on the predicted system states  $\underline{\mathbf{x}}_n$  and the corresponding control inputs  $\underline{\mathbf{u}}_n$ , as well as a terminal reward  $g_N(\underline{\mathbf{x}}_{k+N})$ . The optimal control input  $\underline{\mathbf{u}}_k^*$  of the  $N$  step open-loop optimal control problem is determined according to

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

with

$$V_k(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k) = \max_{\substack{\underline{\mathbf{u}}_{k+1}, \dots, \\ \underline{\mathbf{u}}_{k+N-1}}} \mathbb{E}_{\substack{\underline{\mathbf{x}}_{k+1}, \dots, \\ \underline{\mathbf{x}}_{k+N}}} \left\{ g_N(\underline{\mathbf{x}}_{k+N}) + \sum_{n=k}^{k+N-1} g_n(\underline{\mathbf{x}}_n, \underline{\mathbf{u}}_n) \right\}, \quad (5)$$

which is then applied to the system at time step  $k$ . The whole procedure is repeated in the next time step  $k+1$ .

The analytic evaluation of (5) is typically not possible in case of nonlinear systems. One reason for this is the required prediction of system states for a noise affected nonlinear system. The other one is the necessity to calculate expected values, which also cannot be performed in closed form. Therefore, in the next sections an integrated approach to overcome these two problems is presented.

## III. EFFICIENT STATE PREDICTION

The proposed NMPC framework relies on efficiently calculating the probability density  $\tilde{f}_{k+1}^x(\underline{\mathbf{x}}_{k+1})$  of the system state  $\underline{\mathbf{x}}_{k+1}$  for the next time step  $k+1$ . This prediction is performed by utilizing the so-called Chapman-Kolmogorov equation [1]

$$\tilde{f}_{k+1}^x(\underline{\mathbf{x}}_{k+1}) = \int_{\mathbb{R}^d} \tilde{f}_{\underline{\mathbf{u}}_k}^T(\underline{\mathbf{x}}_{k+1} | \underline{\mathbf{x}}_k) \tilde{f}_k^x(\underline{\mathbf{x}}_k) d\underline{\mathbf{x}}_k, \quad (6)$$

where  $\tilde{f}_{\underline{\mathbf{u}}_k}^T(\underline{\mathbf{x}}_{k+1} | \underline{\mathbf{x}}_k)$  is the transition density, which depends on system (1) and the current control input.<sup>1</sup> Since exactly solving (6) is generally impossible for nonlinear systems, approximate solutions are essential. One very common approach in context of NMPC is linearizing the system and then applying the Kalman filter [9], where the resulting single Gaussian density is typically not sufficient for approximating  $\tilde{f}_{k+1}^x(\underline{\mathbf{x}}_{k+1})$ .

For the scalar system

$$\mathbf{x}_{k+1} = a(\mathbf{x}_k, \underline{\mathbf{u}}_k) + \mathbf{w}_k, \quad (7)$$

with additive stationary white noise, the approach proposed in [8] allows to perform a closed-form and efficient prediction based on a Gaussian mixture approximation of the transition density  $\tilde{f}_{\underline{\mathbf{u}}_k}^T(\mathbf{x}_{k+1} | \mathbf{x}_k) = f^w(\mathbf{x}_{k+1} - a(\mathbf{x}_k, \underline{\mathbf{u}}_k))$  from (6) for every discrete control input  $\underline{\mathbf{u}}_k$ , where  $f^w(\cdot)$  is the noise density. This approximate transition density can be determined off-line, since the nonlinear system is time-invariant and the noise is stationary. Furthermore, it consists of  $L$  Gaussian components, which are axis-aligned, i.e., the covariance matrices of the Gaussian components are diagonal. This special structure allows the analytical evaluation of (6), resulting in a very accurate Gaussian mixture approximation

$$f_{k+1}^x(\underline{\mathbf{x}}_{k+1}) = \sum_{i=1}^L \omega_i \cdot \mathcal{N}(\mathbf{x}_{k+1} - \underline{\mu}_i; \sigma_i^2)$$

<sup>1</sup>Please note that true densities are indicated with a tilde, e.g.  $\tilde{f}(\cdot)$ , while the corresponding approximations are denoted by  $f(\cdot)$ .

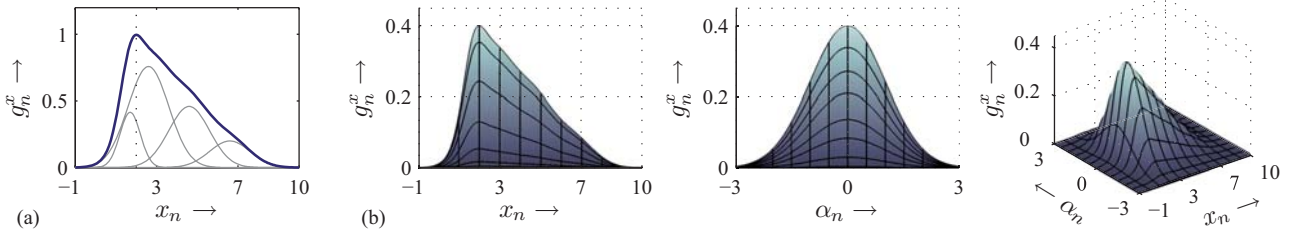


Fig. 1. Asymmetric reward functions, where (a) is an 1-D function with a maximum at  $\tilde{x}_n = 2$  with 4 components (gray lines) and (b) its 2-D extension also comprising the angle (with a maximum at  $\tilde{\alpha}_n = 0$ ).

of the true predicted density  $\tilde{f}_{k+1}^x(x_{k+1})$ . Here,  $L$  is the number of components that is identical to the number of Gaussians of the approximate transition density.  $\omega_i$  are weighting coefficients with  $\omega_i > 0$  and  $\sum_{i=1}^L \omega_i = 1$ .  $\mathcal{N}(x_{k+1} - \mu_i; \sigma_i^2)$  is a Gaussian density with mean  $\mu_i$  and variance  $\sigma_i^2$ .

Even though approximating transition densities can be performed off-line, it is in particular computationally demanding for multi-dimensional system states. To reduce the complexity of approximating the transition density corresponding to system (1) and to enforce an efficient state prediction, the *concept of modularization* is applied [16]. Here, (1) is decomposed into vector-valued subsystems, similar to Rao-Blackwellised particle filters [4].

Recall that we assume that the nonlinear system (1) is corrupted by element-wise additive noise, we can reduce system (1) to a set of less complex subsystems

$$\mathbf{x}_k^{(i+1)} = \underline{a}^{(i)}(\mathbf{x}_k^{(i)}, \underline{u}_k) + \underline{w}_k^{(i)}, \text{ for } i = 1, \dots, m,$$

such that

$$\begin{aligned} \mathbf{x}_{k+1} &= \underline{a}(\mathbf{x}_k, \underline{u}_k, \underline{w}_k) \\ &= \underline{a}^{(m)}(\mathbf{x}_k^{(m)}, \underline{u}_k) + \underline{w}_k^{(m)} \\ \mathbf{x}_k^{(m)} &= \underline{a}^{(m-1)}(\mathbf{x}_k^{(m-1)}, \underline{u}_k) + \underline{w}_k^{(m-1)} \\ &\vdots \\ \mathbf{x}_k^{(2)} &= \underline{a}^{(1)}(\mathbf{x}_k^{(1)}, \underline{u}_k) + \underline{w}_k^{(1)}, \end{aligned}$$

where  $\mathbf{x}_k^{(i)}$  are *auxiliary system states* and  $\underline{w}_k^{(i)}$  are mutually stochastically independent subvectors of the noise  $\underline{w}_k$ . The approximation of the transition densities  $\tilde{f}_{\underline{u}_k}^T(\mathbf{x}_k^{(i+1)} | \mathbf{x}_k^{(i)})$  of each subsystem can be further reduced to the scalar system case (7) (for details see [16]). Finally, given the approximate transition density for each subsystem, calculating the predicted density  $f_{k+1}^x(\mathbf{x}_{k+1})$  can be performed by nested predictions starting with  $\mathbf{x}_k^{(1)} = \mathbf{x}_k$ .

#### Example System: Modularization

The system model (2) describing the mobile robot can be modularized into the subsystems

$$\mathbf{x}_k^{(2)} = \sin(\alpha_k) + \mathbf{w}_k^x \quad (8)$$

and

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + v \cdot T \cdot \mathbf{x}_k^{(2)}, \\ \alpha_{k+1} &= \alpha_k + u_k. \end{aligned} \quad (9)$$

The auxiliary system state  $\mathbf{x}_k^{(2)}$  is stochastically dependent on  $\alpha_k$ . We omit this dependence in further investigations of the example system for simplicity.

#### IV. REWARD FUNCTION REPRESENTATIONS

In this section, a very versatile possibility to model value functions employing Gaussian mixture reward functions is presented, where arbitrary reward functions can be realized due to the Gaussian mixtures' universal approximation property [11]. Obviously, in this case the Gaussian mixtures may have arbitrary parameters, e.g. negative weights  $\omega$ . As already introduced in (4), cumulative value functions  $J_k(\mathbf{x}_k)$  are considered, where  $J_k(\mathbf{x}_k)$  is the maximum achievable expected reward within the next  $N$  steps, which is the sum of the step rewards  $g_n(\mathbf{x}_n, \underline{u}_n)$  and the terminal reward  $g_N(\mathbf{x}_n)$ . For simplicity, step rewards that are additively decomposable according to

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

are considered, although the proposed framework is not limited to this case.

Exploiting the fact that the densities of the predicted state variables are, as explained in the previous section, described by Gaussian mixture densities, the expected value  $E_{\mathbf{x}_n}\{g_n^x(\mathbf{x}_n)\}$  from (5) can be calculated efficiently in closed form, as it is just the multiplication of two Gaussian mixtures with subsequent marginalization [16].

#### Example System: Gaussian Mixture Value Function

If the robot introduced in the previous example is intended to move at a certain optimal distance (e.g.  $\tilde{x}_n = 2$ ) to the wall, where the wall is at  $x_n^{Wall} = 0$  and being closer to the wall is considered less desirable than being farther away, this can be modeled with a reward function as depicted in Fig. 1 (a). If not only the distance to the wall, but also the orientation is to be incorporated in 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. 1 (b). Here, the robot is also driven to move in parallel to the wall.

The input-dependent part of the value function  $g_n^u(\underline{u}_n)$  can either be modeled similar to the procedure described above or with 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 value function representations presented above, (5) can be solved analytically for a finite set of control inputs.

## V. DYNAMIC PROGRAMMING AND VALUE FUNCTION APPROXIMATION

By combining the techniques from Section III and IV, a closed-form calculation of the open-loop optimal control problem within NMPC is possible, where this straightforward solution is an exhaustive tree search with depth  $N$ . In case of very short optimization horizons and a limited number of possible inputs this leads to an efficient and, apart from modeling and approximation errors of the state prediction, exact solution. In case of longer optimization horizons, the straightforward calculation becomes unfeasible as the computational demand increases exponentially with the length  $N$  of the optimization horizon. This problem can be resolved by employing dynamic programming (DP) for calculating the value function (4). This is possible since the value function only consists of additive terms that each just depends on the system state and input for one individual time step [2]. Thus, (4) can be also calculated in a backward fashion according to

$$J_N(\mathbf{x}_{k+N}) = g_N(\mathbf{x}_{k+N}),$$

$$J_n(\mathbf{x}_n) = \max_{\underline{u}_n} \left\{ \underbrace{g_n(\mathbf{x}_n, \underline{u}_n) + \mathbb{E}_{\mathbf{x}_{n+1}} \{J_{n+1}(\mathbf{x}_{n+1}) | \mathbf{x}_n\}}_{V_n(\mathbf{x}_n, \underline{u}_n)} \right\},$$

$$n = k + N - 1, \dots, k.$$

This is highly valuable, as the computational demand now just increases linearly with  $N$ . Additionally, it is now also possible to calculate (5) off-line, which obviously increases the on-line performance significantly.

For calculating  $J_n(\mathbf{x}_n)$ , two consecutive operations have to be executed per step. First,  $V_n(\mathbf{x}_n, \underline{u}_n)$  has to be calculated, where the calculation of the conditional expected value  $\mathbb{E}_{\mathbf{x}_{n+1}} \{J_{n+1}(\mathbf{x}_{n+1}) | \mathbf{x}_n\}$  is the demanding part, and then the value function has to be maximized with respect to  $\underline{u}_n$ .

### A. Calculation of $V_n(\mathbf{x}_n, \underline{u}_n)$

In case the system (1) is represented by a transition density  $f_{\underline{u}_n}^T(\mathbf{x}_{n+1} | \mathbf{x}_n)$  as introduced in Section III, an approximation of  $V_n(\mathbf{x}_n, \underline{u}_n)$  can be calculated by

$$V_n(\mathbf{x}_n, \underline{u}_n) \approx$$

$$g_n(\mathbf{x}_n, \underline{u}_n) + \int_{\mathbb{R}^d} f_{\underline{u}_n}^T(\mathbf{x}_{n+1} | \mathbf{x}_n) \cdot J_{n+1}(\mathbf{x}_{n+1}) d\mathbf{x}_{n+1}$$

$$=: V_n^a(\mathbf{x}_n, \underline{u}_n). \quad (10)$$

If, as it is here the case,  $f_{\underline{u}_n}^T(\mathbf{x}_{n+1} | \mathbf{x}_n)$  and  $J_{n+1}(\mathbf{x}_{n+1})$  are given by Gaussian mixtures, this calculation can be executed very easily and in closed form, as the integral can be solved by just the multiplication of two Gaussian mixtures with a subsequent marginalization. Thus, the resulting function is also a Gaussian mixture.

In the more likely case that the system is described by modularized or nested subsystem as described in Section III, a closed-form solution can be derived similarly.

**Example System: Calculation of  $\mathbb{E}_{\mathbf{x}_{n+1}} \{J_{n+1}(\mathbf{x}_{n+1}) | \mathbf{x}_n\}$**   
With the modularized description of the example system according to (8), (9), where  $v \cdot T = 1$  and (10) it follows

$$\mathbb{E}_{\mathbf{x}_{n+1}} \{J_{n+1}(\mathbf{x}_{n+1}) | \mathbf{x}_n\}$$

$$\approx \int_{\mathbb{R}^2} f_{u_n}^T(x_{n+1}, \alpha_{n+1} | x_n, \alpha_n)$$

$$\cdot J_{n+1}(x_{n+1}, \alpha_{n+1}) dx_{n+1} d\alpha_{n+1}$$

$$= \int_{\mathbb{R}^3} J_{n+1}(x_{n+1}, \alpha_{n+1}) \cdot \delta(x_n + x_n^{(2)} - x_{n+1})$$

$$\cdot \delta(\alpha_n + u_n - \alpha_{n+1}) \cdot f_{\sin}^T(x_n^{(2)} | \alpha_n) dx_n^{(2)} dx_{n+1} d\alpha_{n+1}$$

$$= \int_{\mathbb{R}} J_{n+1}(x_n + x_n^{(2)}, \alpha_n + u_n) \cdot f_{\sin}^T(x_n^{(2)} | \alpha_n) dx_n^{(2)}.$$

Here,  $f_{\sin}^T(x_n^{(2)} | \alpha_n)$  is the axis-aligned Gaussian mixture transition density corresponding to (8) and  $\delta(\alpha_n + u_n - \alpha_{n+1})$  as well as  $\delta(x_n + x_n^{(2)} - x_{n+1})$  are the transition densities corresponding to (9). With

$$J_{n+1}(x_n + x_n^{(2)}, \alpha_n + u_n)$$

$$\approx \sum_{i=1}^L \omega_{i,n+1} \cdot \mathcal{N}\left((x_n + x_n^{(2)}) - \mu_{i,x,n+1}; \sigma_{i,x,n+1}^2\right)$$

$$\cdot \mathcal{N}\left((\alpha_n + u_n) - \mu_{i,\alpha,n+1}; \sigma_{i,\alpha,n+1}^2\right)$$

and

$$f_{\sin}^T(x_n^{(2)} | \alpha_n)$$

$$= \sum_{j=1}^M \omega_j \cdot \mathcal{N}\left(x_n^{(2)} - \mu_{j,x^{(2)}}; \sigma_{j,x^{(2)}}^2\right) \cdot \mathcal{N}\left(\alpha_n - \mu_{j,\alpha}; \sigma_{j,\alpha}^2\right),$$

the calculation, which is a multiplication of two axis-aligned Gaussian mixtures with subsequent integration, can be calculated in closed form analog to the prediction step of the well known Kalman filter. The result is again a Gaussian mixture with  $L \cdot M$  axis-aligned components.

### B. Maximization

The necessary maximization of  $V_n^a(\mathbf{x}_n, \underline{u}_n) \approx V_n(\mathbf{x}_n, \underline{u}_n)$  with respect to  $\underline{u}_n$  is demanding as it is the maximization of Gaussian mixtures comprising components with different weights, means, and covariances. As, to our knowledge, there is no closed-form solution available we propose the following algorithm to find an approximate solution:

- 1) Calculate the sum of  $V_n^a$  for all possible  $\underline{u}_n$ :  
 $V_n^\Sigma := \sum_{\underline{u}_n} V_n^a.$
- 2) Reduce the number of mixture components to a fixed number, e.g. with Salmond's algorithm as described in [15] extended by a feature to generate axis-aligned Gaussian mixtures:  $V_n^{\Sigma, red} := \text{reduce}(V_n^\Sigma).$
- 3) Calculate the maximum of  $V_n^a$  at the mean values of the components of  $V_n^{\Sigma, red}.$
- 4) Generate a new Gaussian mixture  $J_n^a(\mathbf{x}_n)$  consisting of Gaussians with means and covariances as  $V_n^{\Sigma, red}.$  The weights have to be chosen in a fashion, that the new Gaussian mixture takes the true values as calculated in 3). This can be performed by simply solving the resulting linear equation system. The necessary formulas can be found in [12].

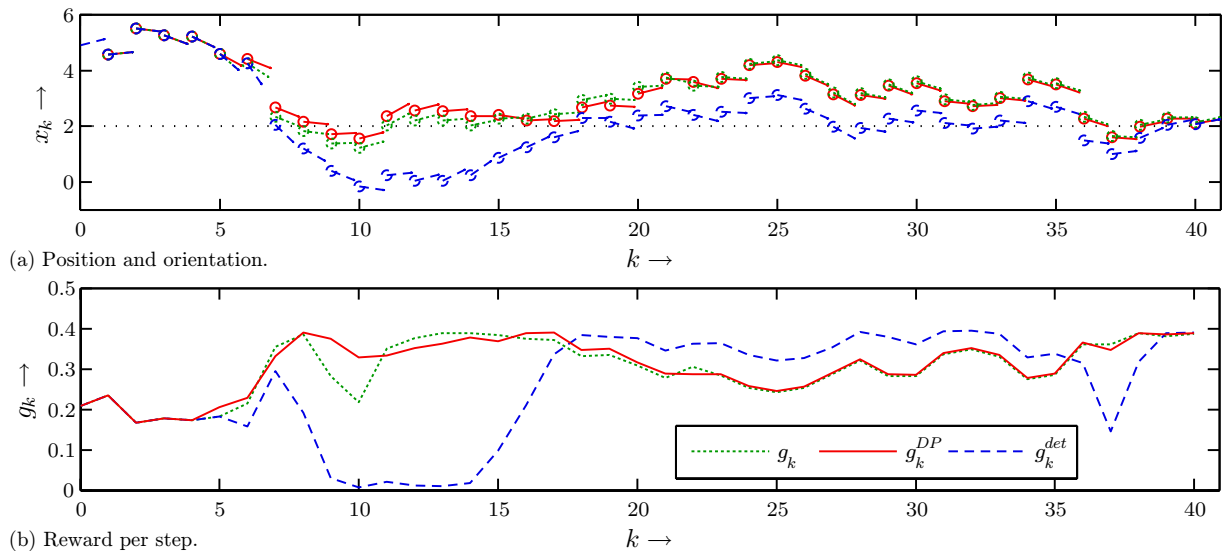


Fig. 2. 40 step simulation. (green dotted line: stochastic NMPC, red solid line: stochastic NMPC with DP, blue dashed line: deterministic NMPC).

The first two steps lead to a discretization of the state space in a meaningful fashion as the mixture reduction algorithm in 2) is intended to keep sufficient support of mixture components at places, where  $V_n^\Sigma$  and therefore the individual  $V_n^a$  are non-zero. Thus, a discretization is achieved that captures the properties of the individual value functions and is therefore not restricted to a certain predefined area of the state space or a predefined arrangement (e.g. a grid) of mixture components, as e.g. proposed in [12]. Based on this discretization, the required maxima can easily be calculated. Please note, that not the reduced, but the original functions are evaluated for this purpose. Finally, a new Gaussian mixture is generated in 4) that has a constant maximum number of components and is axis-aligned. This function can then be directly used as a starting point for the next (backward) step of the dynamic programming procedure.

The algorithm presented in this section permits the efficient calculation of the value function (5) of the open-loop optimization within NMPC. Thus, it is possible to calculate (5) off-line, which obviously leads to an extreme speedup in the on-line control.

## VI. SIMULATIONS

Several simulations have been conducted using the above example scenario to illustrate the modeling capabilities of the proposed framework as well as to emphasize the benefits that can be gained by the direct consideration of noise in the optimal control optimization. Additionally, the dynamic programming approach is compared to the straightforward approach in order to verify its validity.

The considered system is given by (2) and (3), with  $v \cdot T = 1$  and  $u_k \in \{-0.2, 0, 0.2\}$ , where modularization is employed as described above. The considered noise influence on the system  $w_k^x$  is zero-mean white Gaussian noise with standard deviation  $\sigma_w^x = 0.5$ . The measurement noise is also zero-mean white Gaussian noise with standard deviation  $\sigma_v^x = 0.5$  and  $\sigma_v^\alpha = 0.1 \approx 5.7^\circ$ . All simulations are done

for a  $N = 4$  step prediction horizon, with a value function according to (4), where  $g_N(\mathbf{x}_{k+N}, \boldsymbol{\alpha}_{k+N})$  is the function depicted in Fig. 1 (b) and  $g_n(\mathbf{x}_n, \boldsymbol{\alpha}_n) = g_N(\mathbf{x}_{k+N}, \boldsymbol{\alpha}_{k+N})$ .

To evaluate the benefits of the proposed NMPC framework, three different kind of simulations are performed:

1) *Calculation of the input without any consideration of noise (deterministic NMPC)*: The deterministic control is used as a benchmark neglecting the noise influence.

2) *Straightforward calculation of the optimal input considering all noise influences (stochastic NMPC)*: The straightforward calculation of the optimal input with consideration of the noise are carried out using the techniques presented in Section III and Section IV. Thus, it is possible to execute all calculations analytically without the need for any numerical methods. Still, this approach has the drawback that the computational demand for the optimal control problem increases exponentially with the length of the horizon  $N$ . Therefore, this approach is only suitable for short horizons.

3) *Calculation of the optimal input with a value function approximation scheme and dynamic programming (stochastic NMPC with Dynamic Programming)*: In order to be able to use the framework efficiently also for long prediction horizons, it is necessary to employ Dynamic Programming. This is done using the techniques as described in Section V.

In Fig. 2 (a), the first 40 steps of a simulation run are shown. The distance to the wall  $x_k$  is depicted by the position of the circles, the orientation  $\alpha_k$  by the orientation of the arrows. For each simulation run a particular noise realization is used that is applied to the different controllers. Besides that the system is heavily noise influenced 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  $\hat{x}_k = 2$  and  $\hat{\alpha}_k = 0$  while it totally neglects the asymmetry of the reward function. The stochastic control leads to a larger distance from the wall, which is caused by

TABLE I  
SIMULATION RESULTS

	det. control	stoch. control	stoch. DP control
average reward	0.2563	0.2875	0.2897
normalized av. reward	100%	112.16%	113.04%

the non-symmetric reward function together with the noise affecting the system. The two stochastic controllers behave very similar, which illustrates the good performance of the techniques proposed in Section V.

In Fig. 2 (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 start values. The uniformly distributed start values are sampled from the interval  $x_0 \in [0, 6]$  and  $\alpha_0 \in [-\pi/4, \pi/4]$ . In Table I, the average step rewards of the 100 simulations with 40 steps each are shown. To facilitate the comparison also normalized average step rewards are given. Here, it can be seen that the stochastic controllers outperform the deterministic one by over 12% and 13% respectively in terms of reward. In more than 85% of the runs, both stochastic controllers gave overall better results than the deterministic one. Therefore, the performance increase by the stochastic controllers is highly significant with a p-Value of below  $10^{-13}$ . It also can be seen, that the controller using dynamic programming is even slightly better than the straightforward one. This emphasizes the high quality of the proposed value function approximation, especially in comparison to the very basic value function approximation technique used in [16].

## VII. CONCLUSIONS AND FUTURE WORK

A novel framework for closed-form Nonlinear Model Predictive Control for continuous state spaces and a finite set of control inputs that directly incorporates the noise influence in the corresponding optimal control problem has been presented. By using the proposed state prediction methods, which are based on transition density approximations by Gaussian mixture densities and complexity reduction techniques, the otherwise not analytically solvable state prediction of nonlinear noise affected systems can be performed in an efficient closed-form manner. Another very important aspect of NMPC is the modeling of the reward function. The proposed methods that also use Gaussian mixtures lead to a level of flexibility far beyond the traditional representations. By employing the same representation for both the predicted probability density functions and the reward functions, NMPC is solvable in closed form for nonlinear systems with consideration of noise influences. Additionally, an efficient algorithm for calculating the value function of the related optimal control problem is presented that permits its off-line calculation, which significantly reduces the required on-line computational demand and also allows the incorporation of long prediction horizons. The effectiveness of the presented framework and its algorithms as well as the importance of the consideration of noise in the controller has been

shown in simulations of a two-wheeled differential-drive robot following a specified trajectory.

Future research is directed towards the extension to inhomogeneous noise, i.e., noise with state and/or input dependent noise levels, where the incorporation of nonlinear filtering techniques is expected to be highly beneficial. An additional important task will be the consideration of stability aspects, especially in cases of approximated value functions. This can, e.g. be tackled by the use of bounding techniques for the approximation error [10]. Of special interest is the extension to the related emerging field of Model Predictive Sensor Scheduling [7], which is of especial 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, February 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., June 2004.
- [4] N. de Freitas, "Rao-Blackwellised Particle Filtering for Fault Diagnosis," in *IEEE Aerospace Conference Proceedings*, vol. 4, 2002, pp. 1767–1772.
- [5] M. P. Deisenroth, T. Ohtsuka, F. Weissel, D. Brunn, and U. D. Hanebeck, "Finite-Horizon Optimal State Feedback Control of Nonlinear Stochastic Systems Based on a Minimum Principle," in *Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 371–376.
- [6] 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.
- [7] 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, December 2004, pp. 743–748.
- [8] M. Huber, D. Brunn, and U. D. Hanebeck, "Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density," in *Proc. of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006, pp. 98–103.
- [9] J. H. Lee and N. L. Ricker, "Extended Kalman Filter Based Nonlinear Model Predictive Control," in *Industrial & Engineering Chemistry Research*, 1994, pp. 1530–1541.
- [10] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1249–1260, August 2006.
- [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*, June 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. Badgwell, "An Overview of Industrial Model Predictive Control Technology," *Chemical Process Control*, vol. 93, pp. 232–256, 1997.
- [15] D. J. Salmond, "Mixture Reduction Algorithms for Target Tracking in Clutter," in *SPIE Signal and Data Processing of Small Targets*, ser. 1305, April 1990, pp. 434–445.
- [16] F. Weissel, M. F. Huber, and U. D. Hanebeck, "A Closed-Form Model Predictive Control Framework for Nonlinear Noise-Corrupted Systems," in *Proc. of the 4rd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, 2007.