

A CLOSED-FORM MODEL PREDICTIVE CONTROL FRAMEWORK FOR NONLINEAR NOISE-CORRUPTED SYSTEMS

Florian Weissel, Marco F. Huber, Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory, Universität Karlsruhe (TH), Germany
weissel@ieee.org, marco.huber@ieee.org, uwe.hanebeck@ieee.org

Keywords: Nonlinear Model Predictive Control; Stochastic Systems; Nonlinear Estimation

Abstract: In this paper, a framework for Nonlinear Model Predictive Control (NMPC) that explicitly incorporates the noise influence on systems with continuous state spaces is introduced. By the incorporation of noise, which results from uncertainties during model identification and the measurement process, the quality of control can be significantly increased. Since NMPC requires the prediction of system states over a certain horizon, an efficient state prediction technique for nonlinear noise-affected systems is required. This is achieved by using transition densities approximated by axis-aligned Gaussian mixtures together with methods to reduce the computational burden. A versatile cost function representation also employing Gaussian mixtures provides an increased freedom of modeling. Combining the prediction technique with this value function representation allows closed-form calculation of the necessary optimization problems arising from NMPC. The capabilities of the framework and especially the benefits that can be gained by considering the noise in the controller are illustrated by the example of a mobile robot following a given path.

NOTATION

x	variable
\mathbf{x}	random variable
$\underline{\mathbf{x}}$	vector-valued random variable
$\tilde{f}^x(x)$	probability density function of \mathbf{x}
$f^x(x)$	approximate of $\tilde{f}^x(x)$
$\mathcal{N}(x - \mu; \sigma^2)$	Gaussian density with mean μ and standard deviation σ
$E_{\underline{\mathbf{x}}}\{\mathbf{x}\}$	expected value of $\underline{\mathbf{x}}$
$J_k(\underline{\mathbf{x}}_k)$	value function
$V_k(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k)$	input dependent value function
$g_n(\underline{\mathbf{x}}_n, \underline{\mathbf{u}}_n)$	cost function
k	time index
n	time index of prediction horizon

1 INTRODUCTION

Model Predictive Control (MPC), which is also referred to as Receding or Rolling Horizon Control, has become more and more important for control applica-

tions from various fields. This is due to the fact that not only the current system state, but also a model-based prediction of future system states over a finite N stage prediction horizon is considered in the control law. For this prediction horizon, an open-loop optimal control problem with a corresponding value function is solved. Then the resulting optimal control input is applied as a closed-loop control to the system.

The well understood and widely used MPC for linear system models (Qin and Badgwell, 1997) together with linear or quadratic cost functions is not always sufficient if it is necessary to achieve even higher quality control, e.g. in high precision robot control or in the process industry. Steadily growing requirements on the control quality can be met by incorporating nonlinear system models and cost functions in the control. 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 (Findeisen and Allgöwer, 2002) and advances in the employed algorithms to solve the connected open-loop optimization (Ohtsuka, 2003).

Nevertheless, in most approaches, especially for the important case of continuous state spaces, the influence of noise on the system is not considered (Carmacho and Bordons, 2004), which obviously leads to unsatisfactory solutions especially for highly nonlinear systems and cost functions. In (Deisenroth et al., 2006) an extension of the deterministic cost function by a term considering the noise is presented. In (Nikovski and Brand, 2003) 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 as any discretization suffers from the curse of dimensionality.

In technical applications, like robotics or sensor-actuator-networks, discrete-time controllers for systems with continuous-valued state spaces, e.g. the posture of a robot, but a finite set of control inputs, e.g. turn left / right or move straight, are of special importance. Therefore, in this paper a framework for discrete-time NMPC for continuous state spaces and a finite set of control inputs is presented that is based on the efficient state prediction of nonlinear stochastic models. Since an exact density representation in closed form and with constant complexity is preferable, a prediction method is applied that is founded on the approximation of the involved system transition densities by axis-aligned Gaussian mixture densities (Huber et al., 2006). To lower the computational demands for approximating multi-dimensional transition densities, the so-called modularization for complexity reduction purposes is proposed. Thus, the Gaussian mixture representation of the predicted state can be evaluated efficiently with high approximation accuracy. As an additional part of this framework, an extremely flexible representation of the cost function, on which the optimization is based, is presented. Besides the commonly used quadratic deviation, a versatile Gaussian mixture representation of the cost function is introduced. This representation is very expressive due to the universal approximation property of Gaussian mixtures. Combining the efficient state prediction and the different cost function representations, an efficient integrated closed-form approach to NMPC for nonlinear noise affected systems with novel abilities is obtained.

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 3, the efficient closed-form prediction approach for nonlinear systems based on transition density approximation and complexity reduction is derived. Different techniques for modeling the cost function are introduced

in Section 4. In Section 5, three different kinds of NMPC controllers are compared based on simulations employing the example system, which has been introduced in previous sections. Concluding remarks and perspectives on future work are given in Section 6.

2 PROBLEM FORMULATION

The considered discrete-time system is given by

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

where \mathbf{x}_k denotes the vector-valued random variable of the system state, \underline{u}_k the applied control input, and $\underline{a}(\cdot)$ a nonlinear, time-invariant function. \mathbf{w}_k denotes the white stationary noise affecting the system additively *element-wise*, i.e., the elements of \mathbf{w}_k are processed in $\underline{a}(\cdot)$ just additively. For details see Section 3.3.

Example System

A mobile two-wheeled differential-drive robot is supposed to drive along a given trajectory, e.g. along a wall, with constant velocity. This robot can be modeled by the distance to the wall \mathbf{x}_k and its orientation relative to the wall $\boldsymbol{\alpha}_k$, which leads to the discrete-time nonlinear system description

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

where $\mathbf{x}_k = [\mathbf{x}_k, \boldsymbol{\alpha}_k]^\top$, v is a constant velocity, T the sampling interval and \mathbf{w}_k^x and \mathbf{w}_k^α denote 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 \mathbf{y}_k^x and orientation \mathbf{y}_k^α with respect to the wall according to

$$\begin{aligned} \mathbf{y}_k^x &= \mathbf{x}_k + \mathbf{v}_k^x, \\ \mathbf{y}_k^\alpha &= \boldsymbol{\alpha}_k + \mathbf{v}_k^\alpha, \end{aligned} \quad (3)$$

where \mathbf{v}_k^x and \mathbf{v}_k^α describe the measurement noise.

At any time step k , the system state is predicted over a finite N step prediction horizon. Then an open-loop optimal control problem is solved, i.e., the optimal input \underline{u}_k^* is determined according to

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

with

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

where the optimality is defined by a cumulative value function $J_k(\mathbf{x}_k)$

$$J_k(\mathbf{x}_k) = \min_{\underline{u}_k} V_k(\mathbf{x}_k, \underline{u}_k) \quad (5)$$

comprising the step costs $g_n(\mathbf{x}_n, \mathbf{u}_n)$ depending on the predicted system states \mathbf{x}_n and the corresponding control inputs \mathbf{u}_n , as well as a terminal cost $g_N(\mathbf{x}_{k+N})$. This optimal control input \mathbf{u}_k^* is then applied to the system at time step k . In the next time step $k+1$ the whole procedure is repeated.

For most nonlinear systems, the analytical evaluation of (4) is not possible. 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.

3 STATE PREDICTION

Predicting the system state is an important part in NMPC for noise-affected systems. The probability density $\tilde{f}_{k+1}^x(\mathbf{x}_{k+1})$ of the system state \mathbf{x}_{k+1} for the next time step $k+1$ has to be computed utilizing the so-called Chapman-Kolmogorov equation (Schweppe, 1973)

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

The transition density $\tilde{f}_{\mathbf{u}_k}^T(\mathbf{x}_{k+1}|\mathbf{x}_k)$ depends on the system described by (1). For linear systems with Gaussian noise the Kalman filter (Kalman, 1960) provides an exact solution to (6), since this equation is reduced to the evaluation of an integral over a multiplication of two Gaussian densities, which is analytically solvable.

For nonlinear systems, an approximate description of the predicted density $\tilde{f}_{k+1}^x(\mathbf{x}_{k+1})$ is inevitable, since an exact closed-form representation is generally impossible to obtain. One very common approach in context of NMPC is linearizing the system and then applying the Kalman filter (Lee and Ricker, 1994). The resulting single Gaussian density is typically not sufficient for approximating $\tilde{f}_{k+1}^x(\mathbf{x}_{k+1})$. Hence, we propose representing all densities involved in (6) by means of Gaussian mixtures, which can be done efficiently due to their universal approximation property (Maz'ya and Schmidt, 1996).

To reduce the complexity of approximating all density functions corresponding to system (1) and to allow for an efficient state prediction, the *concept of modularization* is proposed, see Section 3.3. Here, (1) is decomposed into vector-valued subsystems. Approximations for these subsystems in turn can be reduced to the scalar case, as stated in Section 3.2. For that purpose, in the following section a

short review on the closed-form prediction approach for scalar systems with additive noise is given. Employing this, modularization enables state prediction for system (1) based on Gaussian mixture approximations of the transition density functions corresponding to scalar systems.

3.1 Scalar Systems

For the scalar system equation

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

the approach proposed by (Huber et al., 2006) allows to perform a closed-form prediction resulting in an approximate Gaussian mixture representation $f_{k+1}^x(x_{k+1})$ of $\tilde{f}_{k+1}^x(x_{k+1})$,

$$f_{k+1}^x(x_{k+1}) = \sum_{i=1}^L \omega_i \cdot \mathcal{N}(x_{k+1} - \mu_i; \sigma_i^2), \quad (7)$$

where L is the number of Gaussian components, $\mathcal{N}(x_{k+1} - \mu_i; \sigma_i^2)$ is a Gaussian density with mean μ_i , standard deviation σ_i , and weighting coefficients ω_i with $\omega_i > 0$ as well as $\sum_{i=1}^L \omega_i = 1$. For obtaining this approximate representation of the true predicted density that provides high accuracy especially with respect to higher-order moments and a multimodal shape, the corresponding transition density $\tilde{f}_{\mathbf{u}_k}^T(x_{k+1}|x_k)$ from (6) is approximated off-line by the Gaussian mixture

$$\begin{aligned} \tilde{f}_{\mathbf{u}_k}^T(x_{k+1}, x_k, \boldsymbol{\eta}) \\ = \sum_{i=1}^L \omega_i \cdot \mathcal{N}(x_{k+1} - \mu_{i,1}; \sigma_{i,1}^2) \cdot \mathcal{N}(x_k - \mu_{i,2}; \sigma_{i,2}^2) \end{aligned}$$

with parameter vector

$$\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T, \dots, \boldsymbol{\eta}_L^T]^T,$$

comprising L axis-aligned Gaussian components (short: axis-aligned Gaussian mixture), i.e., the covariance matrices of the Gaussian components are diagonal, with parameters

$$\boldsymbol{\eta}_i^T = [\omega_i, \mu_{i,1}, \sigma_{i,1}, \mu_{i,2}, \sigma_{i,2}].$$

The axis-aligned structure of the approximate transition density allows performing repeated prediction steps with constant complexity, i.e., a constant number L of mixture components for $f_{k+1}^x(x_{k+1})$.

This efficient prediction approach can be directly applied to vector-valued systems, like (1). However, off-line approximation of the multi-dimensional transition density corresponding to such a system is computationally demanding. Therefore, in the next two sections techniques to lower the computational burden are introduced.

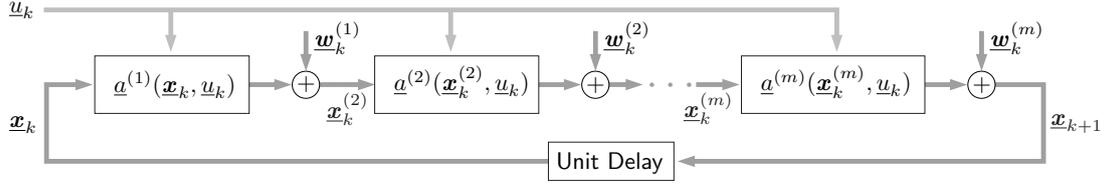


Figure 1: Modularization of the vector valued system $\mathbf{x}_{k+1} = \underline{a}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$.

3.2 Vector-Valued Systems

Now we consider the vector-valued system

$$\mathbf{x}_{k+1} = \underline{a}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \quad (8)$$

with $\mathbf{x}_{k+1} = [\mathbf{x}_{k+1,1}, \mathbf{x}_{k+1,2}, \dots, \mathbf{x}_{k+1,d}]^T \in \mathbb{R}^d$ and noise $\mathbf{w}_k = [\mathbf{w}_{k,1}, \mathbf{w}_{k,2}, \dots, \mathbf{w}_{k,d}]^T \in \mathbb{R}^d$. Assuming \mathbf{w}_k to be white and stationary (but not necessarily Gaussian or zero-mean), with *mutually stochastically independent* elements $\mathbf{w}_{k,j}$, approximating the corresponding transition density $\tilde{f}_{\mathbf{u}_k}^T(\mathbf{x}_{k+1}|\mathbf{x}_k) = \tilde{f}^w(\mathbf{x}_{k+1} - \underline{a}(\mathbf{x}_k, \mathbf{u}_k))$ can be reduced to the scalar system case.

Theorem 1 (Composed Transition Density)

The transition density $\tilde{f}_{\mathbf{u}_k}^T(\mathbf{x}_{k+1}|\mathbf{x}_k)$ of system (8) is composed of separate transition densities of the scalar systems $a_j(\cdot)$, $j = 1, 2, \dots, d$, where $\underline{a}(\cdot) = [a_1(\cdot), a_2(\cdot), \dots, a_d(\cdot)]^T$.

PROOF. Marginalizing $\tilde{f}_{k+1}^x(\mathbf{x}_{k+1})$ from the joint density function $\tilde{f}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{w}_k)$ and separating the elements of \mathbf{w}_k leads to

$$\begin{aligned} \tilde{f}_{k+1}^x(\mathbf{x}_{k+1}) &= \int_{\mathbb{R}^{2d}} \delta(\mathbf{x}_{k+1} - \underline{a}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{w}_k) \tilde{f}_k^x(\mathbf{x}_k) \tilde{f}^w(\mathbf{w}_k) d\mathbf{x}_k d\mathbf{w}_k \\ &= \int_{\mathbb{R}^{2d}} \prod_{j=1}^d \delta(x_{k+1,j} - a_j(\mathbf{x}_k, \mathbf{u}_k) - w_{k,j}) \\ &\quad \cdot \tilde{f}_k^x(\mathbf{x}_k) \prod_{j=1}^d \tilde{f}^{w_j}(w_{k,j}) d\mathbf{x}_k d\mathbf{w}_k \\ &= \int_{\mathbb{R}^d} \left(\prod_{j=1}^d \underbrace{\tilde{f}^{w_j}(x_{k+1,j} - a_j(\mathbf{x}_k, \mathbf{u}_k))}_{\text{separate transition densities}} \right) \tilde{f}_k^x(\mathbf{x}_k) d\mathbf{x}_k. \end{aligned}$$

As a result of the mutually stochastically independence of the elements in \mathbf{w}_k , the transition density of the vector-valued system (8) is separated into d transition densities of d scalar systems. Approximating these lower-dimensional transition densities is possible with decreased computational demand (Huber et al., 2006). \square

The concept of modularization, introduced in the following section, benefits strongly from the result obtained in Theorem 1.

3.3 Concept of Modularization

For our proposed NMPC framework, we assume that the nonlinear system is corrupted by element-wise additive noise. Incorporating this specific noise structure, the previously stated closed-form prediction step can indirectly be utilized for system (1). Similar to Rao-Blackwellised particle filters (de Freitas, 2002), we can reduce the system in (1) to a set of less complex subsystems with a form according to (8),

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

We name this approach *modularization*, where the subsystems

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

correspond to transition densities, that can be approximated according to Section 3.1 and 3.2. Since these subsystems are less complex than the overall system (1), approximating transition densities is also less complex. Furthermore, a nested prediction can be performed to obtain the predicted density $f_{k+1}^x(\mathbf{x}_{k+1})$, see Fig. 1. Starting with $\mathbf{x}_k^{(1)} = \mathbf{x}_k$, each subsystem $\underline{a}^{(i)}(\cdot)$ receives an *auxiliary system state* $\mathbf{x}_k^{(i)}$ and generates an auxiliary predicted system state $\mathbf{x}_k^{(i+1)}$.

The noise \mathbf{w}_k is separated into its subvectors $\mathbf{w}_k^{(i)}$ according to

$$\mathbf{w}_k = [\mathbf{w}_k^{(1)}, \mathbf{w}_k^{(2)}, \dots, \mathbf{w}_k^{(m)}]^T,$$

in case that the single noise subvectors $\mathbf{w}_k^{(i)}$ are mutually independent.

Example System: Modularization

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

$$\mathbf{x}_k^{(2)} = \sin(\boldsymbol{\alpha}_k) + \mathbf{w}_k^x$$

and

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + v \cdot T \cdot \mathbf{x}_k^{(2)}, \\ \boldsymbol{\alpha}_{k+1} &= \boldsymbol{\alpha}_k + u_k + \mathbf{w}_k^\alpha. \end{aligned}$$

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

Please note that there are typically stochastic dependencies between several auxiliary system states. To consider this fact, the relevant auxiliary system states have to be augmented to conserve the dependencies. Thus, the dimensions of these auxiliary states need not all to be equal.

4 COST FUNCTIONS

In this section, two possibilities to model cost functions, the well known quadratic deviation and a novel approach employing Gaussian mixture cost functions, are presented. Exploiting the fact that the predicted state variables are, as explained in the previous section, described by Gaussian mixture densities, the necessary evaluation of the expected values in (4) can be calculated efficiently in closed-form for both options.

In the following, cumulative value functions according to (5) are considered, where $g_n(\mathbf{x}_n, u_n)$ denotes a step cost within the horizon and $g_N(\mathbf{x}_n)$ a cost depending on the terminal state at the end of the horizon. The value function $J_k(\mathbf{x}_k)$ is the minimal cost for the next N steps of the system, starting at state \mathbf{x}_k plus the terminal cost $g_N(\mathbf{x}_n)$.

For simplicity, step costs that are additively decomposable according to

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

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

4.1 Quadratic Cost

One of the most popular cost functions is the quadratic deviation from a target value \check{x} or \check{u} according to

$$g_n^x(\mathbf{x}_n) = (\mathbf{x}_n - \check{x}_n)^\top (\mathbf{x}_n - \check{x}_n).$$

As in our framework the probability density function of the state \mathbf{x}_n is given by an axis-aligned Gaussian mixture $f_n^x(\mathbf{x}_n)$ with L components, the calculation of $E_{\mathbf{x}_n}\{g_n^x(\mathbf{x}_n)\}$, which is necessary to compute

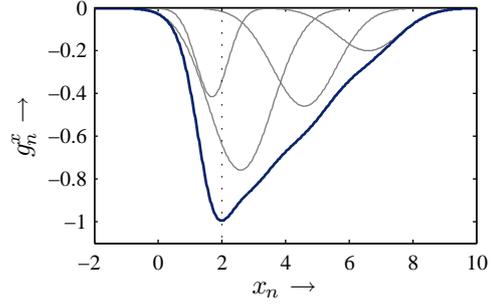


Figure 2: Asymmetric cost function consisting of four components (gray) with a minimum at $\check{x}_n = 2$.

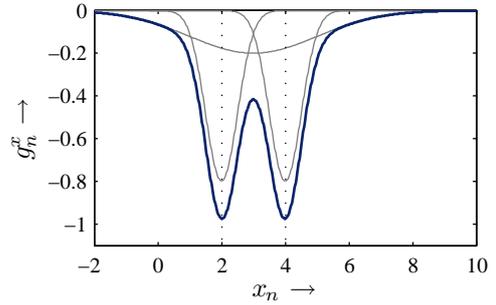


Figure 3: Multimodal cost function consisting of three components (gray) with minima at $\check{x}_n^a = 2, \check{x}_n^b = 4$.

(4), can be performed analytically as it can be interpreted as the sum over shifted and dimension-wise calculated second-order moments

$$\begin{aligned} E_{\mathbf{x}_n}\{g_n^x(\mathbf{x}_n)\} &= E_{\mathbf{x}_n}\{(\mathbf{x}_n - \check{x}_n)^\top (\mathbf{x}_n - \check{x}_n)\} \\ &= \text{trace } E_{\mathbf{x}_n}^2\{(\mathbf{x}_n - \check{x}_n)\} \\ &= \text{trace } \sum_{i=1}^L \omega_i \left((\boldsymbol{\mu}_i - \check{x}_n) (\boldsymbol{\mu}_i - \check{x}_n)^\top + \text{diag}(\boldsymbol{\sigma}_i^2) \right), \end{aligned}$$

employing $E_{\mathbf{x}}^2\{\mathbf{x}\} = \sum_{i=1}^L \omega_i (\boldsymbol{\mu}_i^2 + \boldsymbol{\sigma}_i^2)$.

Example System: Quadratic Cost

If the robot is intended to move parallel along the wall, the negative quadratic deviation of the angle $\boldsymbol{\alpha}_k$ with respect to the wall, i.e., $g_n^\alpha(\boldsymbol{\alpha}_n) = (\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n^{\text{Wall}})^2$ is a suitable cost function.

4.2 Gaussian Mixture Cost

A very versatile description of the cost function can be realized if Gaussian mixtures are employed. In this case, arbitrary cost functions can be realized due to the Gaussian mixtures' universal approximation property (Maz'ya and Schmidt, 1996). Obviously, in this

case the Gaussian mixtures may have arbitrary parameters, e.g. negative weights ω .

Example System: Gaussian Mixture Cost Function

In case the robot is intended to move at a certain optimal distance to the wall (e.g. $\check{x}_n = 2$, with $x_n^{Wall} = 0$), where being closer to the wall is considered less desirable than being farther away, this can, e.g. be modeled with a cost function as depicted in Fig. 2. If two different distances are considered equally optimal, this can be modeled with a cost function as depicted in Fig. 3.

Please note that assigning rewards to a state is equivalent to assigning negative costs, which leads to the depicted negative cost functions in Fig. 2 and Fig. 3.

Here, the calculation of the expected value $E_{\mathbf{x}_n} \{g_n^x(\mathbf{x}_n)\}$, which is necessary for the calculation of (4), can also be performed analytically

$$\begin{aligned} & E_{\mathbf{x}_n} \{g_n^x(\mathbf{x}_n)\} \\ &= \int_{\mathbb{R}^d} f_n^x(\mathbf{x}_n) \cdot g_n^x(\mathbf{x}_n) d\mathbf{x}_n \\ &= \int_{\mathbb{R}^d} \sum_{i=1}^L \omega_i \mathcal{N}(x_n - \underline{\mu}_i; \text{diag}(\underline{\sigma}_i)^2) \\ &\quad \cdot \sum_{j=1}^M \omega_j \mathcal{N}(x_n - \underline{\mu}_j; \text{diag}(\underline{\sigma}_j)^2) d\mathbf{x}_n \\ &= \sum_{i=1}^L \sum_{j=1}^M \omega_{ij} \underbrace{\int_{\mathbb{R}^d} \mathcal{N}(x_n - \underline{\mu}_j; \text{diag}(\underline{\sigma}_{ij})^2) d\mathbf{x}_n}_{=1}, \end{aligned}$$

with

$$\omega_{ij} = \omega_i \omega_j \cdot \mathcal{N}(\underline{\mu}_i - \underline{\mu}_j; \text{diag}(\underline{\sigma}_i)^2 + \text{diag}(\underline{\sigma}_j)^2),$$

where $f_n^x(\mathbf{x}_n)$ denotes the L -component Gaussian mixture probability density function of the system state (7) and $g_n^x(\mathbf{x}_n)$ the cost function, which is a Gaussian mixture with M components.

4.3 Input Dependent Part

The input dependent part of the cost function $g_n^u(u_n)$ can either be modeled similar to the procedures described above or with a lookup-table since there is just a finite number of discrete u_n .

Using the efficient state prediction presented in Section 3 together with the value function representations presented above, (4) can be solved analytically for a finite set of control inputs. Thus, an efficient closed-form solution for the optimal control problem within NMPC is available. Its capabilities will be illustrated by simulations in the next section.

5 SIMULATIONS

Based on the above example scenario, several simulations are conducted to illustrate the modeling capabilities of the proposed framework as well as to illustrate the benefits that can be gained by the direct consideration of noise in the optimal control optimization. The considered system is given by (2) and (3), with $v \cdot T = 1$ and $u_k \in \{-0.2, -0.1, 0, 0.1, 0.2\}$. The considered noise influences on the system \mathbf{w}_k^x and \mathbf{w}_k^α are zero-mean white Gaussian noise with standard deviation $\sigma_w^x = 0.5$ and $\sigma_w^\alpha = 0.05 \approx 2.9^\circ$ respectively. 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 performed for a $N = 4$ step prediction horizon, with a value function according to (5), where $g_N(\mathbf{x}_{k+N})$ is the function depicted in Fig. 2 and $g_n(\mathbf{x}_n, \mathbf{u}_n) = g_N(\mathbf{x}_{k+N}) \forall n$. In addition, the modularization is employed as described above.

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

Calculation of the input without noise consideration (deterministic NMPC):

The deterministic control is calculated as a benchmark neglecting the noise influence.

Direct calculation of the optimal input considering all noise influences (stochastic NMPC):

The direct calculation of the optimal input with consideration of the noise is performed using the techniques presented in the previous sections. 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 . Thus, this approach is suitable for short horizons only.

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 (DP). Unfortunately, this is not directly possible, as no closed-form solution for the value function J_n is available. One easy but either not very accurate or computationally demanding solution would be to discretize the state space. More advanced solutions can be found by value function approximation (Bertsekas, 2000). For the simulations, an especially well-suited case of

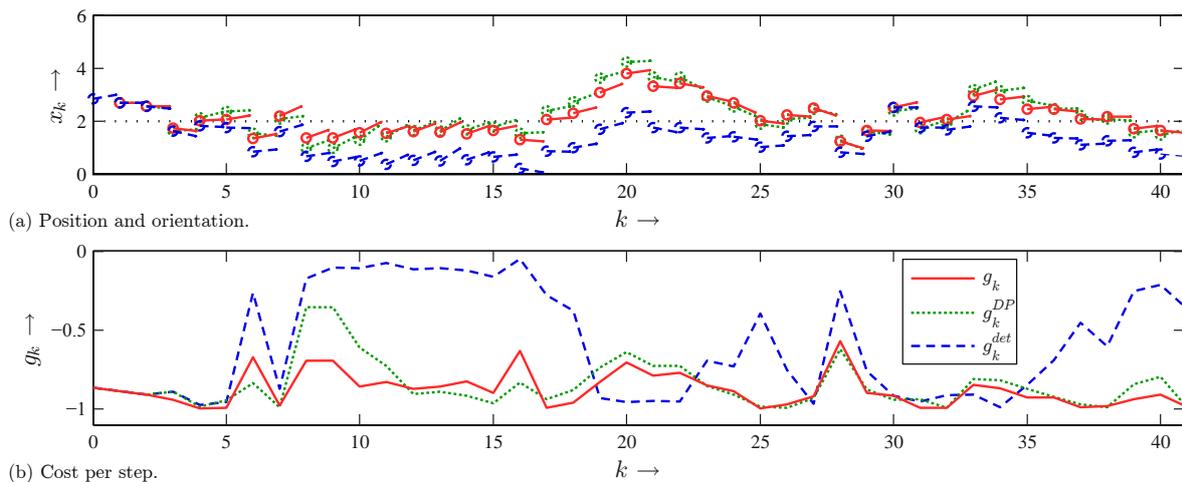


Figure 4: First 40 steps of a simulation (red solid line: stochastic NMPC, green dotted line: stochastic NMPC with DP, blue dashed line: deterministic NMPC).

value function approximation is employed that has been described by (Nikovski and Brand, 2003). Here, the state space is discretized by covering it with a finite set of Gaussians with fixed means and covariances. Then weights, i.e., scaling factors, are selected in such a way that the approximate and the true value function coincide at the means of every Gaussian. Using these approximate value functions together with the techniques described above, again all calculations can be executed analytically. In contrast to the direct calculation, now the computational demand increases only linearly with the length of the prediction horizon but quadratically in the number of Gaussians used to approximate the value function. Here, the value functions are approximated by a total of 833 Gaussians equally spaced over the state space within $(\hat{x}_n, \hat{\alpha}_n) \in \Omega := [-2, 10] \times [-2, 2]$.

For each simulation run, a particular noise realization is used that is applied to the different controllers. In Fig. 4(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 α_k by the orientation of the arrows. Besides that the system is heavily influenced by noise, 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 minimum of the cost function at $\check{x}_k = 2$ and totally neglects the asymmetry of the cost function. The stochastic controllers lead to a larger distance to the wall, as they consider the noise affecting the system in conjunction with the non-symmetric cost function.

In Fig. 4(b), the evaluation of the cost function for each step is shown. As expected, both stochastic controllers perform much better, i.e., they generate less

Table 1: Simulation Results

controller	average cost	
deterministic	-0.6595	(100.00%)
stochastic	-0.7299	(110.66%)
stochastic DP	-0.6824	(103.48%)

cost, 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 interval $x_0 \in [0, 8]$ and $\alpha_0 \in [-\pi/4, \pi/4]$. In Table 1, the average step costs of the 100 simulations with 40 steps each are shown. To facilitate the comparison, also normalized average step costs are given. Here, it can be seen that the stochastic controller outperforms the deterministic one by over 10% in terms of cost. In 82% of the runs, the stochastic controller gives overall better results than the deterministic one. By employing dynamic programming together with value function approximation the benefits are reduced. Here, the deterministic controller is only outperformed by approximately 3.5%. The analysis of the individual simulations leads to the conclusion that the control quality significantly degrades in case the robot attains a state which is less well approximated by the value function approximation as it lies outside Ω . Still, the dynamic programming approach produced better results than the deterministic approach in 69% of the runs. These findings illustrate the need for advanced value function approximation techniques in order to gain the very good control performance of the direct stochastic controller together with the efficient calculation of the DP approach.

6 CONCLUSIONS

A novel framework for closed-form Nonlinear Model Predictive Control (NMPC) for continuous state space and a finite set of control inputs has been presented that directly incorporates the noise influence in the corresponding optimal control problem. By using the proposed state prediction methods, which are based on transition density approximation 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 cost function. The proposed methods also use Gaussian mixtures, which leads to a level of flexibility far beyond the traditional representations. By employing the same representation for both the predicted probability density functions and the cost functions, NMPC is solvable in closed-form for nonlinear systems with consideration of noise influences. The effectiveness of the presented framework and the importance of the consideration of noise in the controller have been shown in simulations of a two-wheeled differential-drive robot following a specified trajectory.

Future research is intended to address various topics. One is the optimization of the value function approximation by abandoning a fixed grid in order to increase performance and accuracy. 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 (Lincoln and Rantzer, 2006). Another interesting extension will be the incorporation of effects of inhomogeneous noise, i.e., noise with state and/or input dependent noise levels. Together with the incorporation of nonlinear filtering techniques this is expected to increase the control quality even more.

Besides the addition of new features to the framework, also the extension to new application fields is intended. Of special interest is the extension of Model Predicted Control to the related emerging field of Model Predictive Sensor Scheduling (He and Chong, 2004), which is of special importance, e.g. in sensor-actuator-networks.

REFERENCES

Bertsekas, D. P. (2000). *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, U.S.A., 2nd edition.

- Camacho, E. F. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag London Ltd., 2 edition.
- Deisenroth, M. P., Ohtsuka, T., Weissel, F., Brunn, D., and Hanebeck, U. D. (2006). Finite-Horizon Optimal State Feedback Control of Nonlinear Stochastic Systems Based on a Minimum Principle. In *Proc. of the IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 371–376.
- Findeisen, R. and Allgöwer, F. (2002). An Introduction to Nonlinear Model Predictive Control. In Scherer, C. and Schumacher, J., editors, *Summerschool on "The Impact of Optimization in Control"*, Dutch Institute of Systems and Control (DISC), pages 3.1–3.45.
- de Freitas, N. (2002). Rao-Blackwellised Particle Filtering for Fault Diagnosis. In *IEEE Aerospace Conference Proceedings*, volume 4, pages 1767–1772.
- He, Y. and Chong, E. K. P. (2004). Sensor Scheduling for Target Tracking in Sensor Networks. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 1, pages 743–748.
- Huber, M., Brunn, D., and Hanebeck, U. D. (2006). Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density. In *Proc. of the IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 98–103.
- Kalman, R. E. (1960). A new Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME, Journal of Basic Engineering*, (82):35–45.
- Lee, J. H. and Ricker, N. L. (1994). Extended Kalman Filter Based Nonlinear Model Predictive Control. In *Industrial & Engineering Chemistry Research*, pages 1530–1541. ACS.
- Lincoln, B. and Rantzer, A. (2006). Relaxing Dynamic Programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260.
- Maz'ya, V. and Schmidt, G. (1996). On Approximate Approximations using Gaussian Kernels. *IMA Journal of Numerical Analysis*, 16(1):13–29.
- Nikovski, D. and Brand, M. (2003). Non-Linear Stochastic Control in Continuous State Spaces by Exact Integration in Bellman's Equations. In *Proc. of the 2003 International Conf. on Automated Planning and Scheduling*, pages 91–95.
- Ohtsuka, T. (2003). A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control. *Automatica*, 40(4):563–574.
- Qin, S. J. and Badgwell, T. A. (1997). An Overview of Industrial Model Predictive Control Technology. *Chemical Process Control*, 93:232–256.
- Schweppe, F. C. (1973). *Uncertain Dynamic Systems*. Prentice-Hall.