

Stochastic Model Predictive Control of Time-Variant Nonlinear Systems with Imperfect State Information

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

Abstract—In many technical systems, the system state, which is to be controlled, is not directly accessible, but has to be estimated from observations. Furthermore, the uncertainties arising from this procedure are typically neglected in the controller. To remedy this deficiency, in this paper, we present a novel approach to stochastic nonlinear model predictive control (NMPC) for heavily noise-affected systems with not directly accessible, i.e., hidden states, extending the stochastic NMPC-framework presented in [1]. An important property of our novel method is that, in contrast to classical approaches, time-variant system and measurement equations as well as time-variant step rewards can be considered. Extending the techniques from [1] by introducing virtual future observations and combining this with a novel tree search algorithm, called probabilistic branch-and-bound search (PBAB), a solution with a feasible computational demand of the challenging problem is possible.

I. INTRODUCTION

In model predictive control (MPC), which is sometimes also referred to as receding or rolling horizon control, the control input is not only determined based on the current system state, but also on an N step prediction.

As the well understood and widely used MPC for linear system models [2] together with 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). Even if the incorporation of nonlinearities leads to a significant increase in the control quality, most approaches do not consider the influence of noise on the system [3], especially in the important continuous state space case. Examples of approaches that do consider the system noise in the control can be found in [4], [5], [6], [7], [8]. However, uncertainties that arise from estimating the system state based on noise corrupted measurements are still not taken into account.

A closely related field, where the influence of system noise and uncertain measurements is considered, are partially observable Markov decision processes (POMDP), where in particular systems with discrete states and discrete observations are considered over an infinite optimization horizon [9], [10]. As even in the discrete case an exact solution is computationally infeasible [11], [12], most approaches solving POMDPs employ iterative numerical methods [13], [14], [15]. To ensure convergence of these methods, a discount factor smaller than one is typically introduced in the reward

function. Existing extensions to the case of continuous states are also based on an iterative procedure in order to approximately solve infinite horizon POMDPs. This requires the system and measurement equation as well as the step reward being time-invariant. For instance, in [16] a discrete POMDP solver is extended to linear systems with continuous states but discrete observations, while [17] employs Monte-Carlo sampling for an approximate solution to POMDPs with continuous states and observations. An approach employing parametric representations of the probability density functions representing the system states to gain an approximate solution is presented in [18].

Of special importance to this paper, besides the cardinality of the state space, is the consideration of not directly accessible system states, i.e., only imperfect state information is available for control. For example, a mobile robot is just able to estimate its pose using a combination of dead-reckoning and observations like distance measurements to landmarks. To gain meaningful estimation results for such systems, the noise significantly affecting the employed models as well as the observations has to be taken into account. Furthermore, as many technical systems are only able to handle a finite set of control inputs even if they have a continuous-valued state space, this paper focuses on these systems. If the aforementioned mobile robot has differential drive kinematics, its state space comprises the continuous-valued pose, but the robot might still just be able to handle commands like turn left / right or move straight.

Based on our stochastic NMPC framework introduced in [1], in this paper an extension to the considerably more demanding setting of stochastic control with imperfect state information is proposed. We now present an approach, where not only the nonlinear system dynamics and the influence of the system noise, but also the nonlinear measurement system together with the measurement noise is considered explicitly in the control by mimicking a recursive estimator within the prediction horizon. For efficient calculation, the techniques from [1], namely the combination of a stochastic state prediction method based on transition density approximation by hybrid densities as well as a matching representation for reward function modeling, which allows closed-form calculation of the necessary reward functions, are extended. Additionally, for efficiently traversing the resulting search tree, a novel algorithm named probabilistic branch-and-bound (PBAB) is introduced.

The remainder of this paper is structured as follows: In the next section, the considered stochastic NMPC problem is described. Our approach for considering uncertainties

F. Weissel, T. Schreiter, M. F. Huber, and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute of Computer Science and Engineering, Universität Karlsruhe (TH), Germany florian.weissel@ieee.org, schreite@ira.uka.de, {marco.huber|uwe.hanebeck}@ieee.org

arising from imperfect state information, which is based on mimicking the state estimator in the controller, is described in Section III. An efficient algorithm for traversing the resulting probabilistic search tree is introduced in Section IV. In Section V, three different kinds of NMPC controllers are compared based on simulations employing an example system from the field of mobile robot control. Conclusions are given in Section VI.

II. PROBLEM FORMULATION

In this paper, discrete-time systems of the form

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

are considered. Here, $\mathbf{x}_k \in \mathcal{X}$ denotes the random vector corresponding to the continuous-valued system state at time step k and \mathbf{u}_k the vector-valued control input from the finite discrete set \mathcal{U}_k . $\mathbf{a}_k(\cdot)$ is a nonlinear, possibly time-variant function, which is affected by additive white noise \mathbf{w}_k , distributed according to the probability density function (PDF) f_k^w , $\mathbf{w}_k \sim f_k^w$. The system state \mathbf{x}_k is not directly accessible, but can only be estimated based on continuous-valued observations $\hat{\mathbf{z}}_k$. The estimation is performed based on the measurement equation

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (2)$$

where $\mathbf{v}_k \sim f_k^v$ is the additive white measurement noise.

Based on the system equation (1) and the measurement equation (2), the stochastic optimal control problem

$$\mathbf{u}_k^*(\mathbf{x}_k) = \arg \max_{\mathbf{u}_{k,0}} \max_{\mathbf{u}_{k,1:N-1}} \underbrace{\sum_{n=1}^N g_n(\mathbf{x}_{k,n})}_{J_{k,1}(\mathbf{x}_{k,1})} \quad (3)$$

is solved for a finite N step horizon at any time step k . Here, the cumulative reward $J_{k,1}(\mathbf{x}_{k,1})$, which comprises step rewards $g_n(\mathbf{x}_{k,n})$, is maximized with respect to the control policy $\mathbf{u}_{k,n}(\mathbf{x}_{k,n})$. This control policy maps a random variable $\mathbf{x}_{k,n} \sim f_{k,n}$, representing the available information on the system state at time n , to a control input $\mathbf{u}_{k,n}^*$. The optimum control input $\mathbf{u}_k^*(\mathbf{x}_k) = \mathbf{u}_{k,0}^*(\mathbf{x}_{k,0})$ is then applied to the system. In the next time step $k+1$, the whole procedure is repeated.

In the considered stochastic setting, the step reward $g_n(\mathbf{x}_{k,n})$ is an operator mapping a random variable $\mathbf{x}_{k,n}$ to a scalar real number. For example, this can be a weighted expected value

$$g_n(\mathbf{x}_{k,n}) = \mathbb{E}_{\mathbf{x}_{k,n}} \{ \hat{g}_n(\mathbf{x}_{k,n}) \},$$

where the weighting function $\hat{g}_n(\mathbf{x}_{k,n})$ is represented by a polynomial or a Gaussian mixture as described in [6]. Alternatively, it can also be a measure representing the uncertainty of a system state like the trace of the covariance matrix.

Generally, the stochastic optimal control problem (3) cannot be solved in closed form for nonlinear systems. Thus, an approximate approach for solving this problem is presented in the following sections.

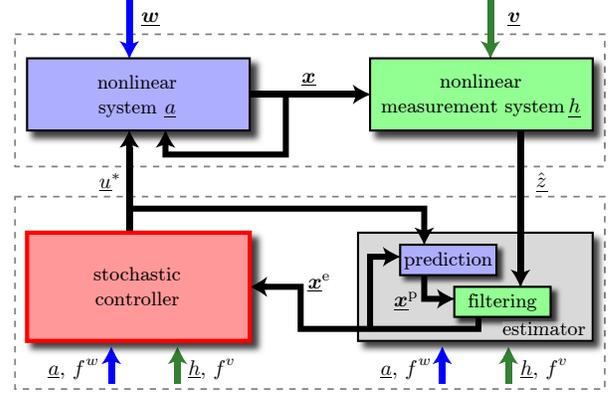


Fig. 1. Stochastically controlled system.

III. CONTROL IN CASE OF IMPERFECT STATE INFORMATION

If the state of a system is not directly accessible, it can be estimated by employing a recursive Bayesian estimator (Fig. 1). The state estimate $\mathbf{x}_k^e \sim f_k^e$ is calculated in two steps:

- 1) In the *prediction step*, the nonlinear system model $\mathbf{a}_k(\mathbf{x}_k, \mathbf{u}_k)$ is used to calculate a one-step prediction $\mathbf{x}_{k+1}^p \sim f_{k+1}^p$ based on the PDF f_k of the current system state \mathbf{x}_k employing the Chapman-Kolmogorov equation [19]

$$f_{k+1}^p(\mathbf{x}_{k+1}) = \int_{\mathcal{X}} f_{\mathbf{u}_k}^T(\mathbf{x}_{k+1}|\mathbf{x}_k) f_k(\mathbf{x}_k) d\mathbf{x}_k, \quad (4)$$

where the transition density $f_{\mathbf{u}_k}^T(\mathbf{x}_{k+1}|\mathbf{x}_k) = f_k^w(\mathbf{x}_{k+1} - \mathbf{a}_k(\mathbf{x}_k, \mathbf{u}_k))$ is the probabilistic representation of the system $\mathbf{a}_k(\mathbf{x}_k, \mathbf{u}_k)$.

- 2) In the *filter step*

$$f_k^e(\mathbf{x}_k) = c_k \cdot f_k^M(\hat{\mathbf{z}}_k|\mathbf{x}_k) \cdot f_k^p(\mathbf{x}_k),$$

an observation $\hat{\mathbf{z}}_k$ is fused with the predicted system state \mathbf{x}_k^p using Bayes' law. Here, c_k is a normalizing constant ensuring that f_k^e is a valid density. The conditional measurement density $f_k^M(\hat{\mathbf{z}}_k|\mathbf{x}_k) = f_k^v(\hat{\mathbf{z}}_k - \mathbf{h}_k(\mathbf{x}_k))$ is the probabilistic representation of the measurement system $\mathbf{h}_k(\mathbf{x}_k)$.

The prediction step needs to be executed at every time step in order to keep the estimate synchronized with the system. The filter step needs to be executed any time a measurement is taken. For clarity of presentation, in the following we consider the case that at every time step exactly one measurement is taken.

The recursive Bayesian estimator provides a general framework for nonlinear system and measurement models affected by noise. However, exact Bayesian estimation in closed form is impossible to achieve in general. Widely-used approximate estimators like the extended Kalman filter [19] and the unscented Kalman filter [20] only provide a Gaussian representation of the resulting densities, which is not sufficient for characterizing these complicated densities.

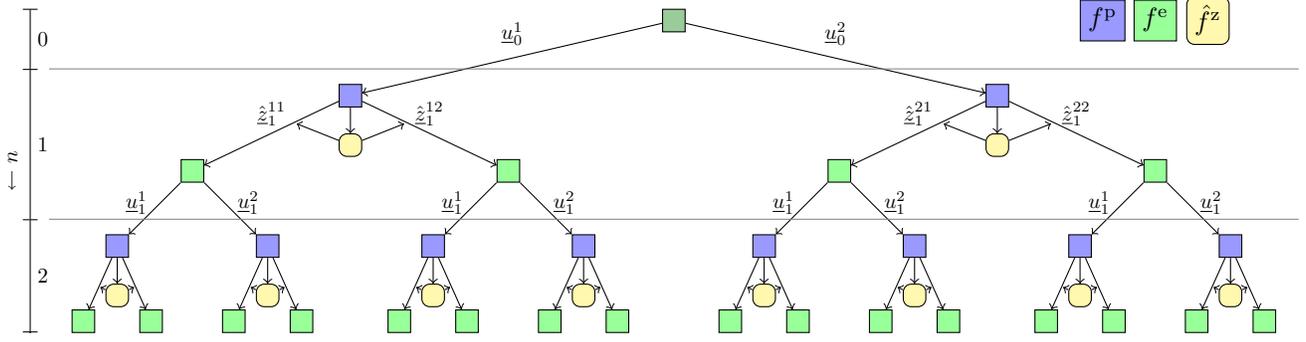


Fig. 2. Search tree for $N = 2$ time steps, $|U| = 2$ different control inputs and $L = 2$ virtual observations with root node f_0 .

Particle filters [21] instead require a large number of samples in order to get satisfactory results, which is computationally demanding. Thus, in this paper the so-called *Hybrid Density Filter (HDF)* [22] is utilized. This estimator offers accurate estimation results thanks to a Gaussian mixture representation of f^p and f^e . Furthermore, estimation can be performed efficiently due to a closed-form approximation of the probabilistic system representations $f_k^M(\underline{z}_k|\underline{x}_k)$ and $f_{\underline{u}_k}^T(\underline{x}_{k+1}|\underline{x}_k)$.

Our approach for solving (3) is to mimic the behavior of the Bayesian estimator within the prediction horizon $n = 1 \dots N$ for any fixed time step k . Thus, the index k is omitted in the following.

Based on the initial PDF f_0 of the system state \underline{x}_0 at time step k , PDFs describing the system states \underline{x}_n at time step n can be calculated for various control inputs $\underline{u}_{0:n-1}$ and observations $\hat{\underline{z}}_{1:n}$ using the HDF. In case of a finite number of *discrete* control inputs and observations, this leads to a tree structure, as depicted in Fig. 2. The control dependent edges from $\underline{x}_n^e \sim f_n^e$ to $\underline{x}_{n+1}^p \sim f_{n+1}^p$ represent a prediction step. The edges from $\underline{x}_n^p \sim f_n^p$ to $\underline{x}_n^e \sim f_n^e$ result from a filter step with $\hat{\underline{z}}_n$.

A. Virtual Observations

As we do not consider discrete, but rather continuous-valued observations, a tree as depicted in Fig. 2 cannot be constructed directly. For determining an optimal solution, the consideration of an infinite number of observations would be required here. To avoid this, an approximate solution employing a finite set of *virtual observations*, which should be as representative as possible, can be used. The term virtual observation is used here, as the true observations are obviously not available at time step $n = 0$. Nevertheless, using the measurement equation (2), the PDF of observations for time steps $n > 0$ can be calculated. This can be achieved in a way similar to the state prediction (4) by propagating the PDF of a system state $f_n^p(\underline{x}_n)$ through the measurement equation (2), which leads to a PDF

$$f_n^z(\underline{z}_n) = \int_{\mathcal{X}} f_n^M(\underline{z}_n|\underline{x}_n) \cdot f_n^p(\underline{x}_n) d\underline{x}_n$$

describing possible future observations. Since an HDF-prediction step is used to perform this calculation, f_n^z is given as a Gaussian mixture.

In order to generate L representative virtual observations, a Dirac mixture approximation

$$f^z(\underline{z}) \approx \hat{f}^z(\underline{z}) = \sum_{l=1}^L \omega^{(l)} \cdot \delta(\underline{z} - \underline{\mu}^{(l)})$$

of the continuous PDF f^z needs to be found. Using the positions $\underline{\mu}^{(l)}$ of the Dirac components as virtual observations, a search tree as depicted in Fig. 2 can then be built up. The weights $\omega^{(l)}$ of the components approximately represent the probability of the individual virtual observations.

The number of employed virtual observations strongly impacts the computational demand of the entire approach. Thus, it is highly desirable to minimize the number of required virtual observations, while preserving a meaningful representation of f^z . In case of one-dimensional observations, the algorithm in [23], which is based on progressively minimizing the Cramér–von Mises distance [24] of $\hat{f}^z(\underline{z})$ and $f^z(\underline{z})$, yields a high quality approximation requiring just a small number of components (Fig. 3). An algorithm, which is also able to approximate multidimensional PDFs, is given in [25].

B. Recursive Calculation

The optimal control input \underline{u}^* can be easily calculated employing (3) if the cumulative reward function $J_1(\underline{x}_1) = J_1(\underline{a}_0(\underline{x}_0, \underline{u}_0) + \underline{w}_0)$ is known. Thus, the main task in determining the control input is calculating the cumulative reward function $J_1(\underline{x}_1)$. Due to the cumulative structure of the reward function, this calculation can be carried out recursively backwards in time. As two different types of

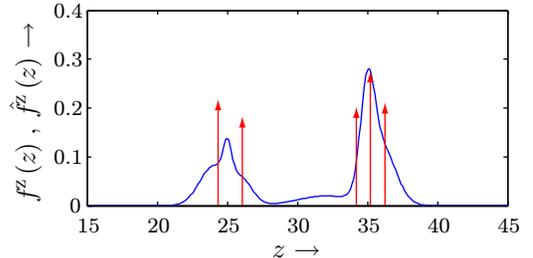


Fig. 3. Virtual observations $\hat{f}^z(\underline{z})$ (red Dirac mixture) approximated based on $f^z(\underline{z})$ (blue) employing the algorithm from [23].

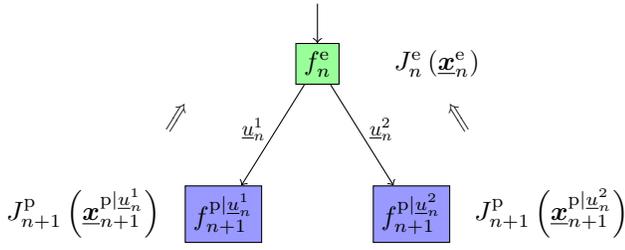


Fig. 4. Calculation of the cumulative reward function $J_n^e(\mathbf{x}_n^e)$.

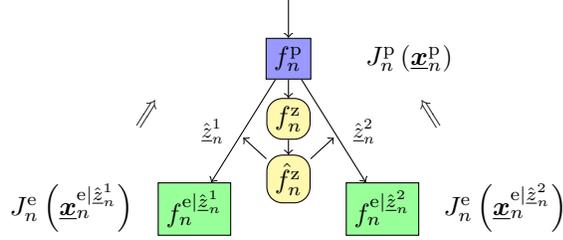


Fig. 5. Calculation of the cumulative reward function $J_n^p(\mathbf{x}_n^p)$.

PDFs, f_n^e and f_n^p , are considered, two different types of value functions, J_n^e and J_n^p , respectively, are also necessary for recursive calculation.

1) *Calculation of $J_n^e(\mathbf{x}_n^e)$* : In the final step of the considered horizon N , the cumulative reward is initialized with the terminal step reward

$$J_N^e(\mathbf{x}_N^e) = g_N(\mathbf{x}_N^e). \quad (5)$$

For $n < N$, any estimated PDF f^e is followed by a predicted PDF f^p as depicted in Fig. 4. The control input that maximizes the total reward is chosen according to (3). Thus, the cumulative reward function $J_n^e(\mathbf{x}_n^e)$ for $n < N$ consists of the sum of the current step reward g_n and the maximum reward of its successors

$$J_n^e(\mathbf{x}_n^e) = g_n(\mathbf{x}_n^e) + \max_{\underline{u}_n} \left\{ J_{n+1}^p(\mathbf{x}_{n+1}^p|\underline{u}_n) \right\}. \quad (6)$$

2) *Calculation of $J_n^p(\mathbf{x}_n^p)$* : The cumulative rewards $J_n^p(\mathbf{x}_n^p)$ can be calculated based on rewards $J_n^e(\mathbf{x}_n^e)$ (Fig. 5), where the PDFs $f_n^e|\hat{z}_n$ of the $\mathbf{x}_n^e|\hat{z}_n$ are dependent on the virtual observations \hat{z}_n . As the occurrence of a certain observation is random, the cumulative reward $J_n^p(\mathbf{x}_n^p)$ is the expected value of its successors $J_n^e(\mathbf{x}_n^e|\hat{z}_n)$ with regard to the virtual observation $\hat{z}_n \sim \hat{f}^z$

$$J_n^p(\mathbf{x}_n^p) = E_{\hat{z}_n} \left\{ J_n^e(\mathbf{x}_n^e|\hat{z}_n) \right\} = \sum_{l=1}^L \omega^{(l)} \cdot J_n^e(\mathbf{x}_n^e|\hat{z}_n^{(l)}). \quad (7)$$

IV. PROBABILISTIC BRANCH-AND-BOUND

A straightforward calculation of the control value, i.e., analyzing the entire search tree, has exponential complexity

$$L^N \cdot |\mathcal{U}|^N \cdot N.$$

Employing more advanced tree search techniques, this computational demand can be significantly reduced while

retaining optimality. In the following, an efficient algorithm based on the well-known branch-and-bound (BAB) algorithm [26] is introduced. The idea of the basic BAB, which is only able to handle deterministic decisions, is to assign an upper bound of the achievable cumulative reward to any visited node. Based on these bounds, the tree is further traversed (branch), while nodes with a higher cumulative reward are considered more promising to lead to the optimal solution and thus are analyzed first.

The basic branch-and-bound algorithm needs to be modified, as, in the considered problem, the tree consists not only of edges that represent control actions, which are deterministic decisions (as in the basic BAB), but also of edges that represent virtual observations for which only a probabilistic representation is available. To consider this, we introduce a novel algorithm (Algorithm 1), the so called *probabilistic branch-and-bound (PBAB)*.

A. Main Routine

In the following, without loss of generality, step reward functions $g_n \leq 0$ are considered. This can easily be achieved by decreasing the reward function by its maximum value.

In the proposed algorithm, a dataset is assigned to every node that contains an upper bound of its cumulative reward \bar{J} , similar to the basic BAB. Additionally, the dataset contains a Boolean flag t indicating whether this upper bound is the true value, i.e., $\bar{J} = J$, and a flag *visited* indicating whether the node has been visited. Initially, \bar{J} are set to 0 (which is always a valid upper bound), and t as well as *visited* are initialized to 0 (indicating FALSE).

The PBAB algorithm (Algorithm 1) starts with an empty search tree except for the root node f_0 . In every cycle of the main loop of the algorithm, the tree is traversed from root to the child node with the best \bar{J} that has not been calculated with certainty, i.e., $t = 0$ (lines 2-4). If this node is not a leaf (line 5), the children \mathbf{C} of this node \mathbf{V} are expanded, and $\mathbf{C}.\bar{J}$ are set to their step-reward g (lines 6-8). This is certainly an upper bound of the true J , as all possible subsequent rewards g are non-positive. $\mathbf{C}.t$ is set to 0 (line 9), as there is no knowledge whether $\mathbf{C}.\bar{J} = \mathbf{C}.J$.

Algorithm 1 Probabilistic Branch-and-Bound

```

1: while  $f_0.t = 0$  do
2:    $\mathbf{V} \leftarrow f_0$ 
3:   while  $\mathbf{V}.visited = 1$  do
4:      $\mathbf{V} \leftarrow$  child of  $\mathbf{V}$  with maximum  $\bar{J}$  and  $t = 0$ 
5:   if  $\mathbf{V} \notin f_n^e$  then
6:      $\mathbf{C} \leftarrow$  EXPANDCHILDREN( $\mathbf{V}$ )
7:     for all children  $\mathbf{C}$  of  $\mathbf{V}$  do
8:        $\mathbf{C}.\bar{J} \leftarrow g(\mathbf{C})$ 
9:        $\mathbf{C}.t \leftarrow 0$ 
10:    UPDATE( $\mathbf{V}$ )
11:   else
12:      $\mathbf{V}.t \leftarrow 1$ 
13:     UPDATE( $\mathbf{V}.parent$ )
14:    $\mathbf{V}.visited \leftarrow 1$ 

```

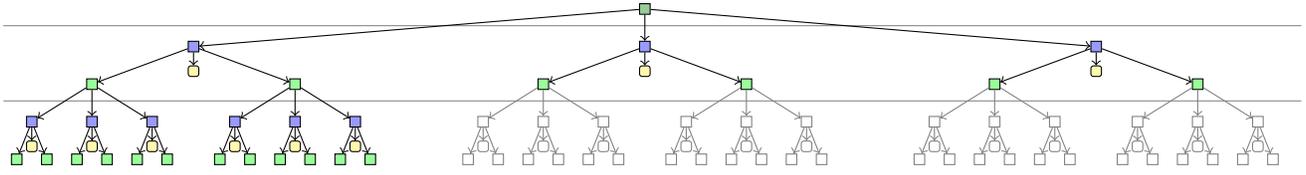


Fig. 6. Search tree where the gray nodes are pruned by PBAB.

Finally, \bar{J} is recursively updated for the current node and its predecessors (line 10), as described in the next paragraph. In case the current node is a leaf node, $\bar{J} = g = J$ (see (5)) and thus $\mathbf{V}.t$ is set to 1 (lines 11-12). In this case \bar{J} is updated recursively starting at the parent of the current node, which is sufficient, as the current node is a leaf-node and thus has no children (line 13). Finally, the *visited* flag of the current node is set to 1.

B. Update

During each iteration of the main loop of PBAB, one new node is analyzed. Following this step, the upper bounds of the cumulative reward \bar{J} are updated recursively as described in Algorithm 2. If the current node is the root node or an f^e node, its children result from a control-dependent prediction step. Thus, the cumulative reward can be updated according to (6) (lines 3-4). Furthermore, the newly calculated \bar{J} is not only an upper bound of J , but the exact value if the maximum \bar{J} of the children is also exact (line 5). Employing this allows the calculation of the exact cumulative reward without analyzing all subsequent subtrees. If the current node is an f^p node, its children result from virtual observations. Thus, the cumulative reward can be updated according to (7) (lines 6-7). Here, the newly calculated \bar{J} is only the exact value, if the \bar{J} of all the children are exact, all these \bar{J} are considered in the calculation (line 8). As long as the current node is not the root, the \bar{J} are recursively updated (line 9).

The PBAB terminates when the cumulative reward J is calculated, i.e., $f_0.\bar{J}$ contains an exact value, i.e., $f_0.t = 1$ (Algorithm 1, line 1). This ensures that the optimal solution always is found.

In the best case, one f^p -node's children have a significantly higher step reward than the f^p node's siblings' children. In this case, always only one subtree needs to be analyzed, which leads to a tree as depicted in Fig. 6. In this

case, the computational complexity

$$L^N \cdot |\mathcal{U}| \cdot N$$

is not exponential in the number of control inputs $|\mathcal{U}|$.

In the special case $L = 1$, where only one virtual observation is employed, the number of computed nodes is even smaller in the best-case, which leads to the complexity of $|\mathcal{U}| \cdot N$.

V. SIMULATIONS

The effectiveness of the proposed approach and especially the benefits that can be gained by considering not only the stochastic system behavior but also the effects of estimating the system state from noise corrupted observations is illustrated by an example from mobile robot control.

A miniature walking robot is supposed to reach a goal through an environment, which is modeled based on a step reward as depicted in Fig. 7. The goal region is modeled with high rewards (light) using Gaussians with a maximum value of 1. Obstacles, e.g., walls, are assigned a penalty of -2 (dark).

The considered mobile robot is a miniature walking robot as described in [27]. This omni-directional robot is used in a mode of operation, where left and right turns are superimposed onto the forward motion. This motion can be modeled similar to the motion of a two-wheeled differential-drive robot, according to the nonlinear discrete-time system equation

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

where the state vector $\underline{\mathbf{x}}_k = [\mathbf{x}_k, \mathbf{y}_k, \phi_k]^\top$ comprises the robot's pose. The control vector $\underline{\mathbf{u}}_k = [u_k^s, u_k^\phi]^\top$ comprises the desired forward step width u_k^s and rotation angle u_k^ϕ . Here, a set of seven control actions

$$\mathcal{U} = \left\{ \begin{bmatrix} 8 \\ \pm 45^\circ \end{bmatrix}, \begin{bmatrix} 8 \\ \pm 22.5^\circ \end{bmatrix}, \begin{bmatrix} 8 \\ 0^\circ \end{bmatrix}, \begin{bmatrix} 0 \\ \pm 45^\circ \end{bmatrix} \right\}$$

is considered. The additive white Gaussian system noise $\underline{\mathbf{w}}_k = [\mathbf{w}_k^x, \mathbf{w}_k^y, \mathbf{w}_k^\phi]^\top$ of the robot has standard deviations $\sigma^x = \sigma^y = 1.5$ and $\sigma^\phi = 3^\circ$, respectively.

Furthermore, the robot is equipped with a sensor to measure the distance

$$z_k = \sqrt{(\mathbf{x}_k - \check{\mathbf{x}}_k)^2 + (\mathbf{y}_k - \check{\mathbf{y}}_k)^2} + \mathbf{v}_k$$

Algorithm 2 Update

```

1: function UPDATE( $\mathbf{V}$ )
2:    $\mathbf{C} \leftarrow$  children of  $\mathbf{V}$ 
3:   if  $\mathbf{V} \in \{f_0, f^e\}$  then
4:      $\mathbf{V}.\bar{J} \leftarrow \max\{\mathbf{C}.\bar{J}\} + g(\mathbf{V})$ 
5:      $\mathbf{V}.t \leftarrow (\arg \max\{\mathbf{C}.\bar{J}\}).t$ 
6:   if  $\mathbf{V} \in f^p$  then
7:      $\mathbf{V}.\bar{J} \leftarrow \mathbb{E}\{\mathbf{C}.\bar{J}\}$ 
8:      $\mathbf{V}.t \leftarrow \min\{\mathbf{C}.t\}$ 
9:   if  $\mathbf{V} \neq f_0$  then
10:    UPDATE( $\mathbf{V}.\text{parent}$ )

```

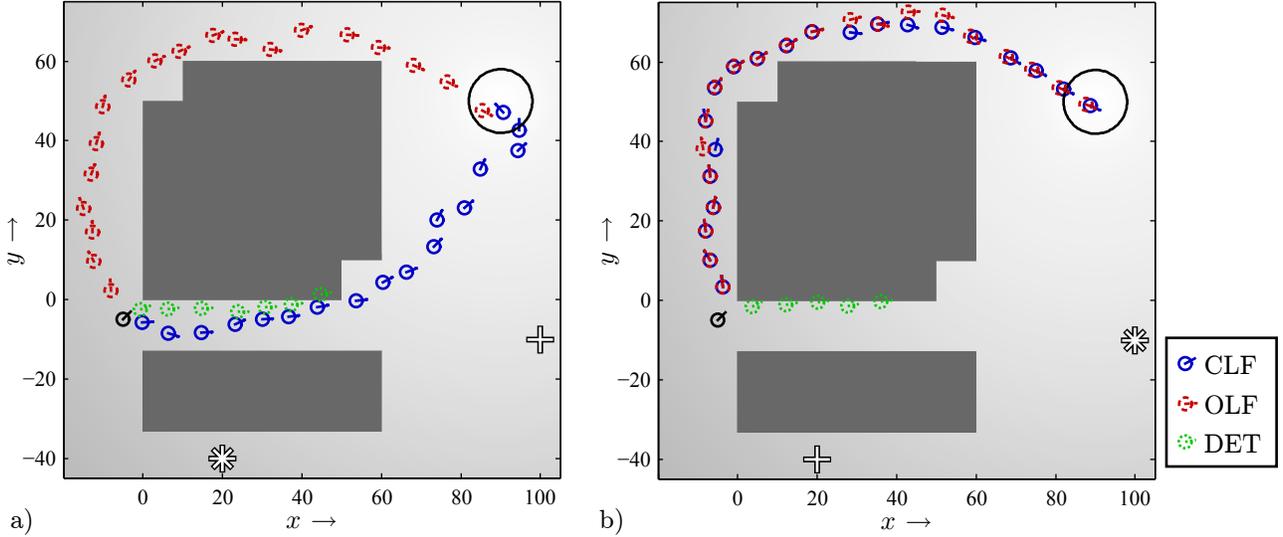


Fig. 7. Simulation with primary landmarks for distance measurement at a) $\tilde{x} = 20$, $\tilde{y} = -40$ and b) $\tilde{x} = 100$, $\tilde{y} = -10$.

to a landmark at $[\tilde{x}_k, \tilde{y}_k]^T$, where v_k denotes the additive white Gaussian measurement noise with standard deviation $\sigma^v = 0.25$.

Two scenarios with identical step reward are considered, which only differ in the the measurements that are taken. The robot, which starts at $\underline{x}_0 = [-5, -5, 45^\circ]^T$, has basically two choices to reach the goal region at $[90, 50]^T$: (1) the left path, which is wide but longer, or (2) the right path, which is shorter but more demanding due to the walls on both sides. In scenario a), there is a primary landmark at $[20, -40]^T$, which is used every time step k , and a secondary landmark at $[100, -10]^T$, which is used in addition every fourth time step. In scenario b), the primary and secondary landmarks switch positions. In the first scenario (Fig. 7 a)), the primary landmark at $[20, -40]^T$ leads to high quality position estimates of the robot in y -direction, which is especially important moving through the narrow part of the right path. In the second scenario (Fig. 7 b)), the primary landmark at $[100, -10]^T$ leads to high quality position estimates in x -direction. The position estimates in y -direction are substantially more uncertain than in the first scenario.

For each of these scenarios, 100 simulation runs with three different kinds of NMPC controllers with an $N = 3$ step prediction horizon are performed, where equal noise realizations are used for all controllers. The simulation stops if the robot hits an obstacle.

Next to the controller employing the presented techniques (CLF), two benchmark controllers are employed. The first one is an Open-Loop Feedback (OLF) SNMPC controller as described in [1], which considers the stochastic system behavior but not the observations in the control. Within the optimization horizon, the assumption is made that no measurements are taken, which leads to conservative approximations.

The second benchmark controller is a deterministic NMPC controller (DET), neglecting any noise influence in the control. To ensure that the robot using this controller does

not hit an obstacle instantly, the penalty region is enlarged by 2 length units for this controller.

In Fig. 7, an example simulation run for each scenario is shown. Independent of the landmark configuration, the robot under deterministic control enters the shorter right path and hits the obstacle, as it neglects any probabilistic information. Again independent of the landmark configuration, the robot using the stochastic OLF controller always uses the wider but longer left path, as it overestimates the uncertainties. The robot employing the proposed Closed-Loop Feedback controller (CLF), which considers imperfect state information and measurements in the control, behaves differently in both scenarios. In scenario a), the distance measurements provide high quality estimates in y -direction and thus, the shorter right path can safely be taken, which is done by the robot. In scenario b), the distance measurements do not provide a sufficient estimation quality in y -direction and, thus, the safer path is selected.

In Table I, results of the 100 simulation runs are shown. It can be seen that both stochastic controllers (CLF and OLF) safely reach the goal region in the vast majority of cases ($\geq 96\%$), contrary to the deterministic one ($\leq 50\%$). The proposed CLF controller chooses in scenario a) the short way and thus only needs an average of 15.9 steps to reach the goal region compared to 20.7 steps required by the OLF controller. The better performance of the CLF controller can also be seen regarding the average reward within the first 25 simulation steps, where only those simulations are considered in this statistics, where the robot does not hit an obstacle.

In the considered setting with an $N = 3$ step prediction horizon and $L = 1$ virtual observations, a search tree with 722 nodes needs to be evaluated. Employing the proposed PBAB algorithm, the necessary computational demand is reduced by a factor of about 8 in the considered scenarios. For longer prediction horizons or an increased number of control inputs, this factor typically even increases.

TABLE I
SIMULATION RESULTS FOR 100 SIMULATION RUNS EACH.

| | a) Primary landmark at $[20, -40]^T$ | | | b) Primary landmark at $[100, -10]^T$ | | |
|--------------------------------------|--------------------------------------|------|------|---------------------------------------|------|------|
| | CLF | OLF | DET | CLF | OLF | DET |
| goal region reached | 98% | 100% | 50% | 96% | 98% | 46% |
| steps needed to reach goal region | 15.9 | 20.7 | 15.1 | 20.6 | 20.7 | 15.2 |
| average step reward (first 25 steps) | 20.7 | 18.4 | 19.9 | 18.6 | 18.5 | 18.7 |

VI. CONCLUSIONS

In this paper, an approach to stochastic NMPC in the case of imperfect state information, i.e., hidden states, is presented. Here, especially systems with a continuous-valued state and observation space as well as a finite set of discrete control inputs are considered. Contrary to other approaches especially from the field of partially observable Markov decision processes, also time-variant system and measurement equations as well as time-variant reward functions can easily be employed. The proposed approach is an extension of our stochastic NMPC framework introduced in [1]. The influence of the uncertainties arising from noise-corrupted measurements is introduced in the control by mimicking the behavior of a Bayesian estimator, which processes so called virtual observations. To reduce the computational burden, an efficient tree search algorithm, probabilistic branch-and-bound (PBAB), is introduced. Combining PBAB with the extended techniques from [1], namely stochastic state estimation based on hybrid densities as well as reward function modeling employing matching representations, a feasible solution of the demanding problem is presented.

REFERENCES

- [1] F. Weissel, M. F. Huber, and U. D. Hanebeck, "A Nonlinear Model Predictive Control Framework Approximating Noise Corrupted Systems with Hybrid Transition Densities," in *Proceedings of the 2007 IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, Louisiana, Dec. 2007, pp. 3661–3666.
- [2] S. J. Qin and T. A. Badgwell, "An Overview of Industrial Model Predictive Control Technology," *Chemical Process Control*, vol. 93, no. 316, pp. 232–256, 1997.
- [3] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Springer-Verlag London Ltd., Jun. 2004.
- [4] 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.
- [5] J. Marecki, S. Koenig, and M. Tambe, "A Fast Analytical Algorithm for Solving Markov Decision Processes with Real-Valued Resources," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Jan. 2007.
- [6] F. Weissel, M. F. Huber, and U. D. Hanebeck, "A Closed-Form Model Predictive Control Framework for Nonlinear Noise-Corrupted Systems," in *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007)*, vol. SPSMC, Angers, France, May 2007, pp. 62–69.
- [7] 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, pp. 3664–3671.
- [8] 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, New York, Jul. 2007, pp. 3751–3756.
- [9] E. B. Dynkin, "Controlled Random Sequences," *Theory of Probability and its Applications*, vol. 10, no. 1, pp. 1–14, 1965.
- [10] R. D. Smallwood and E. J. Sondik, "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, Sep. – Oct. 1973.
- [11] C. Papadimitriou and J. N. Tsitsiklis, "The Complexity of Markov Decision Processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, Aug. 1987.
- [12] O. Madani, S. Hanks, and A. Condon, "On the Undecidability of Probabilistic Planning and Infinite Horizon Partially Observable Markov Decision Problems," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI)*, 1999, pp. 541–548.
- [13] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [14] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, Aug. 2003, pp. 1025–1032.
- [15] M. T. J. Spaan and N. Vlassis, "Perseus: Randomized Point-based Value Iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, Aug. 2005.
- [16] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-Based Value Iteration for Continuous POMDPs," *The Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, Nov. 2006.
- [17] S. Thrun, "Monte Carlo POMDPs," in *Advances in Neural Information Processing Systems 12*, T. L. S.A. Solla and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 1064–1070.
- [18] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte, "Parametric POMDPs for Planning in Continuous State Spaces," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 887–897, 2006.
- [19] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*, 1st ed. Wiley & Sons, 2006.
- [20] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *International Symposium on Aerospace/Defence Sensing, Simulation and Control*, 1997.
- [21] 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.
- [22] 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.
- [23] O. C. Schrempf, D. Brunn, and U. D. Hanebeck, "Dirac Mixture Density Approximation Based on Minimization of the Weighted Cramér-von Mises Distance," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Heidelberg, Germany, Sep. 2006, pp. 512–517.
- [24] D. D. Boos, "Minimum Distance Estimators for Location and Goodness of Fit," *Journal of the American Statistical Association*, vol. 76, no. 375, pp. 663–670, 1981.
- [25] V. Klumpp and U. D. Hanebeck, "Dirac Mixture Trees for Fast Suboptimal Multi-Dimensional Density Approximation," in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Korea, Aug. 2008.
- [26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2003.
- [27] F. Weissel, M. F. Huber, and U. D. Hanebeck, "Test-Environment based on a Team of Miniature Walking Robots for Evaluation of Collaborative Control Methods," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, San Diego, California, Nov. 2007, pp. 2474–2479.