Stochastic Nonlinear Open-Loop Feedback Control with Guaranteed Error Bounds Using Compactly Supported Wavelets

Achim Hekler, Martin Kiefel, and Uwe D. Hanebeck

***Abstract*— In model predictive control, a high quality of control can only be achieved if the model of the system reflects the real-world process as precisely as possible. Therefore, the controller should be capable of both handling a nonlinear system description and systematically incorporating uncertainties affecting the system. Since stochastic nonlinear model predictive control (SNMPC) problems in general cannot be solved in closed form, either the system model or the occurring densities have to be approximated. In this paper, we present an SNMPC framework that approximates the densities and the reward function by their wavelet expansions. Due to the few requirements on the shape and family of the densities or reward function, the presented technique can be applied to a large class of SNMPC problems. For accelerating the optimization, we additionally present an efficient technique, so-called dynamic thresholding, which neglects insignificant coefficients, while at the same time guaranteeing that the optimal control input is still obtained. The capabilities of the proposed approach are demonstrated by simulations and comparisons to a particle-based SNMPC method are conducted.**

I. INTRODUCTION

A. Motivation

In many control applications, it is mandatory to achieve a high quality of control. For example, there are chemical processes, where certain conditions have to be guaranteed in order to avoid damages with serious consequences.

Model predictive control (MPC) is capable of making high-quality decisions. The basic idea of this technique is to predict the current system state over a planning horizon by means of a model of the system. Then, the predicted system states and the corresponding sequence of control inputs are rated by a reward function describing the desired behavior [1].

Certainly, it is important that the model of the system reflects the real-world process as adequately as possible. Therefore, the controller should be capable to deal with nonlinear system models in order to characterize the system more accurately.

But even with the power of nonlinear models, it is in general impossible to model every detail of the real-world process, because of uncertainties in state estimation, exogenous influences, or modeling errors. For incorporation of these unknown quantities, a probabilistic descriptions of both the system state as well as the state transitions are employed. This leads to *stochastic nonlinear model predictive control* (SNMPC) with states described by probability density functions and transitions covered by conditional probability distributions.

A. Hekler and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. Martin Kiefel is with the Max Planck Institute for Intelligent Systems, Tübingen, Germany. Achim.Hekler@kit.edu, martin.kiefel@tuebingen.mpg.de, uwe.hanebeck@ieee.org

B. Related Work

Most of the established control methods do not consider the probabilistic nature of the system under control and work with expected values instead of entire densities [2], [3]. Although such techniques usually have a much lower computational burden, the results can never be as accurate as they could be with a systematic incorporation of the uncertainties. However, SNMPC problems cannot be solved in closed form for general systems and arbitrary densities.

There are two established approaches to cope with this problem. On the one hand, the system model can be simplified, e.g., by linearization. In the case of Gaussian noise and a quadratic reward function, the SNMPC equations can then be solved analytically by an LQG controller [1]. However, especially for strongly nonlinear systems, it is apparent that the arising linearization error may lead to an insufficient approximation of the predicted states and consequently, the quality of control decreases.

On the other hand, the occurring probability densities can be approximated by specific density representations, so that the SNMPC equations can then be solved in closed form. Particularly for the special case of Gaussian noise, it is suitable to employ a Gaussian mixture approximation [4], [5], which is a weighted sum of Gaussian densities. In contrast to the previously mentioned approaches, particle-based SNMPC methods can deal with both strongly nonlinear system as well as arbitrary noise distributions [6], [7]. These approaches approximate the probability densities by a collection of randomly chosen samples, the so-called particles. Instead of propagating the whole probability distribution, only these particles are propagated through the system model. A disadvantage of particle-based methods is that typical distance measures cannot be applied for determining the similarity between two discrete representation or between a continuous and a discrete representation [8].

C. Contribution

In this paper, we present an SNMPC framework for systems with continuous-valued state space and a finite set of control inputs. This class of system is represented in many technical systems. For example, the state space of a walking robot introduced in [9] is its continuous position. In addition,

this kind of robot is only able to handle a finite set of discrete control inputs, namely commands such as turn left/right or move forwards.

The key idea of our approach is to approximate the occurring densities by their wavelet expansions. The advantages of wavelets lie in their good approximation qualities for smooth signals and the ability to describe both the locality and the influencing frequencies at the same time. Therefore, state densities are good candidates for the wavelet transformation, as they are usually localized on a small but essential part in the state space. Furthermore, since wavelets do not make great demands on the shape and family of the densities or reward function, the presented technique can be applied to a large class of SNMPC problems. In contrast to other approximation techniques, e.g., based on Gaussian mixtures, it is even possible to determine the impact of the approximation error on the decision making.

D. Outline of the Paper

The paper is structured as follows: In the next section, a precise definition of the considered problem class is given. Sec. III presents an SNMPC framework solely working with wavelet approximations. In order to achieve a fast control, Sec. IV introduces a novel method for neglecting insignificant coefficients, the so-called *dynamic thresholding*. Finally, conclusions and future work close the paper.

E. General Notation

Throughout this paper, random variables $\boldsymbol{x}$ are written in bold face letters, whereas deterministic quantities x are in normal letters. In order to identify vector-valued quantities $\underline{\boldsymbol{x}}$ or $\underline{x}$, we underline the corresponding identifier. The notation $\underline{\boldsymbol{x}}_k \sim f_k^x(\underline{x}_k)$ means that the random variable $\underline{\boldsymbol{x}}_k$ is characterized by its probability distribution $f_k^x(\underline{x}_k)$.

II. PROBLEM FORMULATION

We assume that the system under control is described by a time-invariant discrete-time stochastic system model

$$\underline{a} : \mathbb{R}^N \times \mathcal{U} \times \mathbb{R}^P \to \mathbb{R}^N \ ,$$

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

where $\underline{a}$ denotes the nonlinear system function and $\underline{u}_k$ the deterministic control input. We assume that the system state $\underline{\boldsymbol{x}}_k$ is partially inaccessible, i.e., sensor information does not uniquely identify the complete state, e.g., due to measurements that are noisy or provide only information about parts of the system state. In this case, the complete system state $\underline{\boldsymbol{x}}_k$ has to be estimated by means of these measurements. The random variable $\underline{\boldsymbol{w}}_k$ subsumes the system noise and can be *arbitrarily* distributed. Throughout this paper, we assume that the system state $\underline{\boldsymbol{x}}_k$ is restricted to a bounded region and the control input $\underline{u}_k$ only takes values from a finite set $\mathcal{U}$.

For the rest of the paper, we work with the probabilistic description of the system

$$f^T(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k) \ ,$$

the so-called transition density, instead of the generative model (1). The transition density can be derived from (1) and the probability density function $f(\underline{w}_k)$ of the system noise $\underline{\boldsymbol{w}}_k$.

The application-specific reward function

$$r : \mathbb{R}^N \times \mathcal{U} \to \mathbb{R} \ ,$$

$$\hat{r} = r(\underline{x}_k, \underline{u}_k) \tag{2}$$

rates, how desirable a system state $\underline{x}_k$ and a control input $\underline{u}_k$ are. Due to the assumed partial inaccessibility of the system state, the reward function has not to rate a deterministic state $\underline{x}_k$, but a state estimate $\underline{\boldsymbol{x}}_k \sim f_k(\underline{x}_k)$. A stochastic extension of (2) can be realized by calculating the expected reward

$$\hat{r} = E\{r(\underline{\boldsymbol{x}}_k, \hat{\underline{u}}_k)\} = \int r(\underline{x}_k, \hat{\underline{u}}_k) f_k(\underline{x}_k) d\underline{x}_k \tag{3}$$

for a given control input $\hat{\underline{u}}_k$ [1].

If we want to rate a whole sequence of sequentially applied control inputs

$$\underline{u}_{k,0:N-1} := \left(\underline{u}_{k,0}, \underline{u}_{k,1}, \ldots, \underline{u}_{k,N-1}\right)$$

and the corresponding predicted system estimates

$$\underline{\boldsymbol{x}}_{k,0:N} := \left(\underline{\boldsymbol{x}}_{k,0}, \underline{\boldsymbol{x}}_{k,1}, \ldots, \underline{\boldsymbol{x}}_{k,N}\right) \ ,$$

we further assume the overall reward function over a planning horizon of length N to be cumulative over time, so that

$$r(\underline{\boldsymbol{x}}_{k,0:N}, \underline{u}_{k,0:N-1}) = \sum_{n=0}^{N-1} r(\underline{\boldsymbol{x}}_{k,n}, \underline{u}_{k,n}) + r(\underline{\boldsymbol{x}}_{k,N}) \ . \tag{4}$$

Throughout this paper, we consider open-loop feedback controllers, which assume no additional information to become available in future steps within the planning horizon. In this case, the objective of the model predictive controller is to maximize the cumulative reward of (4) over all possible sequences of control inputs. The sequence of control inputs

$$u^*_{k,0:N-1} = \arg \max_{u_{k,0:N-1}} E\{\sum_{n=0}^{N-1} r(\underline{\boldsymbol{x}}_{k,n}, \underline{u}_{k,n}) + r(\underline{\boldsymbol{x}}_{k,N})\}$$

that maximizes the expected reward solves the optimization problem of finding the best control input for the system.

Due to the finite set of control inputs, the calculation of the optimal control input can be expressed by a decision tree that is defined through the different choices that we have at every time step. At any given depth of the tree, the controller has to decide on the input that maximizes the reward in the next step. Fig. 1 shows an example decision tree.

The nodes of the tree are the predicted densities f^p, which are dependent on specific control inputs. They can be determined by extrapolating the current state estimate f_k^e over time by the Chapman-Kolmogorov equation

$$f_k^p(\underline{x}_{k+1}) = \int f^T(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k) f_k^e(\underline{x}_k) d\underline{x}_k \tag{5}$$

with $f_{k+1}^e(\underline{x}_{k+1}) = f_k^p(\underline{x}_{k+1})$.

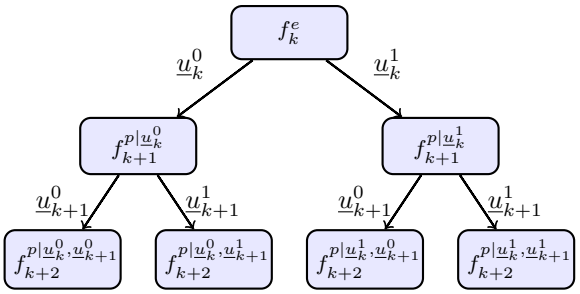

Fig. 1. Simple example of a decision tree for open-loop feedback SNMPC with horizon of length 2. The current state estimate is denoted by f_k^e and only two possible control inputs $\underline{u}^0$ and $\underline{u}^1$ are considered. The resulting predicted densities f^p are rated by the application-specific reward function.

III. SNMPC USING WAVELET APPROXIMATIONS

In this section, the application of wavelets to SNMPC is shown.

We make three simplifying assumptions to increase the readability of the proofs and theorems. First, we assume that the system state is one-dimensional. The extension to the multi-dimensional case can be carried out due to the properties of a separable basis. Second, the control input is implicitly given by the transition density and, hence, will be ignored. A different input at time step t can be simulated by exchanging the system transition density at this time step. Third, to simplify the wavelet expansion of transition densities, we sometimes subsume both the scale and shift parameter in one single index when not stated differently. The set Ω is used to denote all valid combinations.

A. Compactly Supported Wavelets

A wavelet is a function ψ that has a good localization in both time and space and builds a whole family of functions that can be defined with

$$\psi_{j,s}(x) = 2^{-j/2}\psi(2^{-j}x - s) \ .$$

Here, the parameter j is shrinking the *mother wavelet* ψ with powers of 2, whereas s is a shift. Please note that the shifts are getting smaller when the scale is getting finer, i.e., with decreasing j.

In order to understand the following, it is also necessary to know that the wavelet function has M vanishing moments, such that

$$\int x^m \psi(x)dx = 0 \text{ for } m = 0, \cdots, M-1 \ .$$

For a short overview of multiscale analysis and the extension to the multi-dimensional case, we refer the reader to our previous work [10]. More detailed information can also be found in [11], [12].

B. Probabilistic Prediction Using Compactly Supported Wavelets

For the prediction step of a Bayesian estimator, the counterpart in the wavelet domain was derived in [10], whose time complexity is linear with respect to the non-zero coefficients. In particular, the prediction step can be calculated in an exact way and thus, in combination with the approximation quality, a good performance is achieved.

Remark 1 *In the case of time-invariant systems, the coefficients of the transition density can be computed offline. This precalculation reduces the online computation time significantly.*

C. Evaluation of Expected Reward

If both the predicted densities and the reward function are given in their wavelet representations, the calculation of (3) boils down to a sum of coefficients.

Theorem 1 *Let $c_i^{(p)}$ and $c_i^{(r)}$ be the wavelet coefficients of the predicted density f_k^p and the reward function r, respectively. Then, the operation needed to evaluate eq. (3) is given by*

$$\hat{r} = \sum_i c_i^{(p)} c_i^{(r)} \ . \tag{6}$$

Proof: The proof is very similar to that of Theorem 1 in [10]. ■

IV. THRESHOLDING METHOD FOR EFFICIENT WAVELET-BASED SNMPC

In general, a considerable amount of wavelet coefficients is zero or very small due to the vanishing moments and the small support of the used wavelets. In particular, the more the investigated function behaves like a polynomial, the more coefficients are equal or close to zero. Hence, the optimization problem can be considerably simplified by leaving out summands that are not significant during calculation as neglecting them implies only a small error. At the same time, we should always guarantee that the optimal control input is still obtained.

A. State-of-the-Art Thresholding Techniques

There are two established approaches for coefficient thresholding: *hard thresholding* and *soft thresholding* [11]. Hard thresholding discards every coefficient that is smaller than a given value completely independent of its shift or scale. Soft thresholding works in the same way, but additionally decreases every coefficient that is not neglected by the same threshold towards zero. Both thresholding techniques have the property that they decrease the number of coefficients in such a way that the L^2-error relative to the original function remains small. However, in a model predictive setting, this objective is certainly not the right choice. Instead, we should minimize the error introduced by the prediction operation. The following sections explain how a set of coefficients can be found that can safely be ignored while giving a guarantee on the error bound for the prediction operation.

B. Dynamic Thresholding

Here, we propose to set the threshold differently for each scale in order to achieve better results and error guarantees. This technique is what we call *dynamic thresholding*. The idea behind dynamic thresholding is that we suspect higher thresholding constants on finer scales than on coarser ones. Seen in the context of the prediction operation, this means that fine scales may be thresholded more rigorously than coarser ones while inducing the same error.

The next theorem gives a precise answer why detail coefficients fall to zero when increasing the granularity of the approximation.

Theorem 2 *If $f \in L^2(\mathbb{R})$ is uniformly Lipschitz $\alpha < n$ over the interval $[0,1]$, where n is the amount of vanishing moments of the wavelet used, then there exists $A > 0$ such that*

$$|d_{j,s}| \leq A j^{\alpha+\frac{1}{2}} \text{ with } j \in \{\cdots, -1, 0\} \ .$$

Proof: A proof of the theorem can be found in [11]. ∎

We restrict ourselves to a more specific setup, where the aim is the reduction of complexity for evaluating the expected reward for a given state. We would like to select certain coefficients in the representation of the utility function, neglect them, and guarantee that the error, made later by the calculation of the expected reward, is lower than a given error bound. Due to the finite set of control inputs and the given error bound, we can always guarantee that the optimal control input is selected, by propagating the error along the prediction tree.

Let us assume that $\widetilde{r}$ is the correct reward value of a given state and $\hat{r}$ is our modified result of the evaluation operation. Furthermore, let $c_i^{(p)}$ and $c_i^{(r)}$ be the wavelet coefficients of the state and the reward function, respectively, and Ω represents the set of the neglected coefficients. Then, the error regarding the reward is calculated by

$$\begin{aligned} |\widetilde{r} - \hat{r}| &= |\sum_i c_i^{(p)} \cdot c_i^{(r)} - \sum_{i \notin \Omega} c_i^{(p)} \cdot c_i^{(r)}| \\ &= |\sum_{i \in \Omega} c_i^{(p)} \cdot c_i^{(r)}| \ . \end{aligned}$$

Moreover, let us assume we know the largest coefficient amplitude $C_j^{(p)}$ that exists on an arbitrarily chosen scale j, i.e., $C_j^{(p)} = \max_s |c_{j,s}^{(p)}|$, and that Ω_j is the shift index set of neglected coefficients of this scale j. Then, we are able to express the absolute error with

$$\begin{aligned} |\widetilde{r} - \hat{r}| &= |\sum_j \sum_{s \in \Omega_j} c_{j,s}^{(p)} \cdot c_{j,s}^{(r)}| \\ &\leq \sum_j \sum_{s \in \Omega_j} |c_{j,s}^{(p)} \cdot c_{j,s}^{(r)}| \\ &\leq \sum_j C_j^{(p)} \cdot \sum_{s \in \Omega_j} |c_{j,s}^{(r)}| \ . \end{aligned}$$

The overall absolute error is the sum of all absolute errors of the different scales.

Algorithm 1 Dynamic Thresholding

$C \leftarrow \{\max_s c[j,s] | j \in \mathbb{N}\}$
$D_{\text{weighted}} \leftarrow \{(C^{(p)}[j] \cdot c^{(r)}[j,s], (j,s)) | j \in \mathbb{N}, k \in \mathbb{Z}\}$
$D_{\text{sorted}} \leftarrow$ sort D_{weighted}
$k \leftarrow 0$
$(c', (j,s)) \leftarrow$ pop D_{sorted}
while $k + c' < e$ **do**
  $k \leftarrow k + c'$
  $c^{(r)}[j,s] \leftarrow 0$
  $(c', (j,s)) \leftarrow$ pop D_{sorted}
end while

An algorithm that selects the largest possible set of coefficients and sets them to zero is shown in algorithm 1. It sorts all of the coefficients according to their weights $C_j^{(p)} \cdot |c_{j,s}^{(r)}|$ before neglecting the smallest elements until their sum has reached the error bound given by the user.

Now the question is, how large the maximum scale amplitudes $C_j^{(p)}$ for the unknown future state densities are. The answer could be given by experimental evaluation, as similar state densities at the root of the prediction tree lead to similar predicted state densities. By running a set of predictions, we get a pretty good idea of how these constants look like. Note that the constants do not depend on the shift but only on the scale.

Interestingly, if the wrong maximum was chosen, the controller can detect that there has been a false assumption by checking the maxima during the calculation of the predicted state density function. Thus, there is no silent error. In those situations, a fallback to the ordinary calculation yields the right result. The practicability of this approach was tested by means of simulations in the next section.

Moreover, another observation can be made. For many system classes, the state densities usually become more and more *washed-out* during the prediction over the horizon, as no concrete measurement are incorporated into the estimates. This fact also decreases the influence of high frequencies leading to smaller maximum amplitudes on finer scales. A fast open-loop controller can make use of this by using differently thresholded versions of the reward function. If the system becomes more imprecise when getting close to the end of the control horizon, a reward function with fewer coefficients can be used without introducing a higher calculation error. Human planning is very similar to this. Fine details next to some obstacle or landmark far away will unlikely affect the way we plan paths to or around it, unless we are already very close to it.

Certainly, there are systems that may be constructed, where the state estimation keeps the same precision and does not wash out. Here, dynamic thresholding is not meaningful in the progressive way it is introduced in the last paragraph.

Knowing the error of the reward enables a worst-case search, too. In the case where expected rewards overlap through their error intervals, the calculation can be repeated by using the original reward instead.

V. Comparison to Particle-Based SNMPC Methods

Since the assumptions of a particle-based SNMPC approach and the proposed method are very similar, we would like to theoretically compare the two methods, before we present the simulation results in the next section.

As mentioned before, a disadvantage of particle-based methods is that typical distance measures cannot be applied for determining the similarity between two density representation. Consequently, it is generally not possible to determine guaranteed error bounds as it is proposed in this paper.

Another drawback of the particle-based SNMPC is that the complexity increases strongly with the complexity of the reward function. In detail, if the evaluation of the value function for a single particle is computationally complex, the SNMPC problem becomes intractable. In contrast, if the reward function is given in its wavelet expansion, the calculation of the expected reward is a sum of products of the coefficients (see Theorem 1). Additionally, particle-based prediction methods can be arbitrarily bad due to their non-determinism. The proposed method does not suffer from this flaw.

However, a disadvantage of the wavelet-based SNMPC method is that the transition density and the reward function have to be also approximated by means of their wavelet expansions. This transformation can be complex during runtime. But if the considered system is time-invariant as assumed in this paper, the determination of the wavelet coefficients can be performed offline.

VI. Simulations

In order to demonstrate the capabilities of the proposed approach, we conducted several simulation runs. In the chosen simulation scenario, a mobile robot should move along a wall at a safe distance without crashing into the wall.

A. System and Measurement Model

The system state in the experiments is given by $[\boldsymbol{x}_k,\ \boldsymbol{\phi}_k]^T$, where $\boldsymbol{x}_k$ is the distance to the wall and $\boldsymbol{\phi}_k$ the orientation of the robot relative to the wall. We assume that the robot can only execute forward and turning movements similar to a car. Hence, its motion can be modeled according to the following system equation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \sin(\boldsymbol{\phi}_{k+1}) + \boldsymbol{w}_k^{(x)} \ ,$$
$$\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k + \alpha_k + \boldsymbol{w}_k^{(\phi)} \ ,$$

where α_k denotes the control input at time step k and $\boldsymbol{w}_k^{(x)}, \boldsymbol{w}_k^{(\phi)}$ subsume the noise. The noise is assumed to be normally distributed with standard deviations $\sigma_k^{(x)} = 0.5$ and $\sigma_k^{(\phi)} = 0.05$ and is stochastically independent from the system state. As the state space is bounded for the controller presented in this work, only the region $x_k \in [-3, 7]$ and $\phi_k \in [-\pi, \pi]$ is evaluated. For control inputs, only values from the set $\mathcal{U} = \{-0.2, 0, 0.2\}$ are allowed for turning the vehicle at every point in time. In order to improve the state estimation, the robot additionally takes measurements in every time step. The chosen measurement model is given by

$$\underline{\boldsymbol{z}}_k^{(x)} = \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k^{(x)} \ ,$$
$$\underline{\boldsymbol{z}}_k^{(\phi)} = \underline{\boldsymbol{\phi}}_k + \underline{\boldsymbol{v}}_k^{(\phi)} \ ,$$

where the measurement noise is subsumed by the variables $\underline{\boldsymbol{v}}_k^{(x)}, \underline{\boldsymbol{v}}_k^{(\phi)}$ and is distributed in the same way as the system noise.

time step	4	3	2	1
coefficient count	583	526	613	785

TABLE I
Size of reward function after dynamic thresholding

	Ø number of wavelet coeff.	Ø number of particles
f_k^p	649.8	1800
f_k^e	1727.2	1800

TABLE II
Average size of the predicted densities f_k^p and the densities f_k^e after the filter step over the 450 Monte Carlo simulation runs.

B. Reward Function

As mentioned before, the reward function models the desired behavior of the system. Thus, an adequate reward function for our simulation setting gives a small reward if the mobile robot is too close or too far away from the wall. Since the robot should additionally move parallel to the wall, we further punish those states, where the orientation of the robot is pointing strongly towards the wall or away from it. The reward function

$$r(x_k, \phi_k) = \frac{3}{2\pi \cdot 0.5 \cdot \frac{\pi}{2}} \cdot \exp\left(-\frac{1}{2}\left(\left(\frac{x_k - 1.5}{0.5}\right)^2 + \left(\frac{\phi_k}{\frac{\pi}{2}}\right)^2\right)\right) + \operatorname{sign}(x_k) - 1$$

realizes such a behavior. Here, we model the wall to be along $x_k = 0$.

C. Results

We conducted 450 Monte Carlo simulation runs, where each run consists of 50 steps. For approximating the occurring densities, we use Daubechies wavelets with two vanishing moments.

A run with 15 steps and a planning horizon of four steps with an OLF controller was used to collect the density data to create a representative candidate for dynamic thresholding. In Fig. 2, the coefficient amplitude maximum for the different scales is shown given the distance from the current point in time. The scale axis should be read rather as a nominal axis as we used a lexicographical order. The experiments

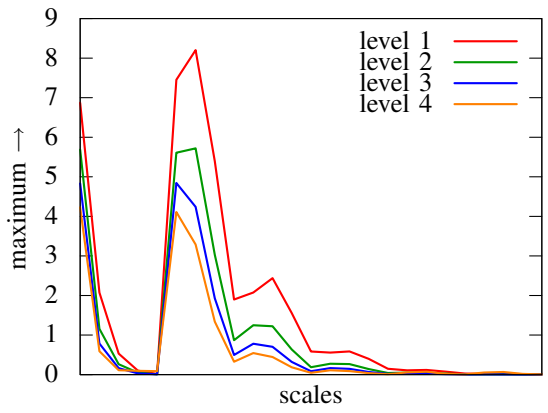

Fig. 2. Amplitude maximum of different scales during the prediction of the OLF controller. The scales axis should be read as a nominal axis as we used a lexicographical order. Multi-dimensional scales are compared according to their coarsest base function component and finer scales are more on the right side than coarser ones. Different time steps within the planning phase are depicted with different symbols.

emphasize the two theoretical result. First, the detail coefficients fall to zero when the granularity of the approximation is increased. Second, the influence of fine scales decreases with increasing level of planning time step.

Table I shows the size of the reward function on the different levels during the prediction. Due to the noise, the simulation suffers from outliers and, hence, the number of coefficients does not fall all the time. The accepted error used as input for the dynamic thresholding algorithm was set to 0.001.

To further emphasize the benefit of the proposed approach, we compare it to a particle-based OLF controller [13]. Both OLF controllers can deal with strongly nonlinear models and arbitrary noise densities. In each test run, the same noise realizations are used for both methods. To measure the quality of a simulation run, we determine the reward of the actual system states in each of the 50 time steps and cumulate these values over time. The wavelet expansion of the transition density was truncated with a hard threshold of 0.01 to get comparable sizes of the density estimations. Although this introduces an additional error that may grow over time during the simulation, we were able to beat the particle-based method.

The proposed method achieved an average cumulative reward of 11.25, while the particle-based approach only reached an average cumulative reward of 8.80. This emphasizes that the proposed method achieves high quality results.

The occuring complexity of the two applied methods can be seen from Table II. While we use an instance of a particle filter with a constant number of 1800 particles, the average number of wavelet coefficients is smaller, namely 649.79 for the predicted density and 1727.21 for the posterior density after the filter step.

VII. CONCLUSIONS AND FUTURE WORK

We presented an SNMPC framework operating on wavelet expansions with constant complexity over time. For accelerating the optimization, a technique for neglecting coefficients was presented, the so-called dynamic thresholding. It gives a precise way to find coefficients that may be discarded without loosing the guarantee of selecting the optimal control input.

The complexity of the proposed control technique does not lie in the computational complexity, but usually in the amount of data that has to be processed. Therefore, we think that it is possible to successfully implement a parallelized version of the shown algorithm. Even an implementation that uses specialized hardware such as FPGAs or graphic chips would be conceivable. Future work might also investigate closed-loop feedback control. Here, the characteristics of the measurement process and of the future disturbances are incorporated in the control law [14]. This might lead to considerably better results in search for an optimal control sequence. Here, wavelets might be a good tool for approximating densities and finding representative measurements.

REFERENCES

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

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

[3] J. Rawlings, "Tutorial overview of model predictive control," *Control Systems Magazine, IEEE*, vol. 20, no. 3, pp. 38–52, 2000.

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

[5] 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.

[6] S. Thrun, "Monte Carlo POMDPs," in *Advances in Neural Information Processing Systems 12*, S. Solla, T. Leen, and K.-R. Müller, Eds. MIT Press, 2000, pp. 1064–1070.

[7] L. Blackmore and B. C. Williams, "Optimal, Robust Predictive Control of Nonlinear Systems under Probabilistic Uncertainty Using Particles," in *Proceedings of the American Control Conference (ACC 2007)*, vol. 17, no. 2, New York, USA, Jul. 2007.

[8] U. D. Hanebeck and O. C. Schrempf, "Greedy Algorithms for Dirac Mixture Approximation of Arbitrary Probability Density Functions," in *Proceedings of the 2007 IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, Louisiana, Dec. 2007, pp. 3065–3071.

[9] 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.

[10] A. Hekler, M. Kiefel, and U. D. Hanebeck, "Nonlinear Bayesian Estimation with Compactly Supported Wavelets," in *Proceedings of the 2010 IEEE Conference on Decision and Control (CDC 2010)*, Atlanta, Georgia, Dec. 2010.

[11] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.

[12] I. Daubechies, *Ten Lectures on Wavelets*, ser. CBMS-NSF Reg. Conf. Series in Applied Math. SIAM, 1992.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Massachusetts: The MIT Press, 2006.

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