2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Stochastic Nonlinear Model Predictive Control Using Gaussian Processes

Eric Bradford[1] and Lars Imsland[2]

Abstract— **Model predictive control is a popular control approach for multivariable systems with important process constraints. The presence of significant stochastic uncertainties can however lead to closed-loop performance and infeasibility issues. A remedy is given by stochastic model predictive control, which exploits the probability distributions of the uncertainties to formulate probabilistic constraints and objectives. For nonlinear systems the difficulty of propagating stochastic uncertainties is a major obstacle for online implementations. In this paper we propose to use Gaussian processes to obtain a tractable framework for handling nonlinear optimal control problems with Gaussian parametric uncertainties. It is shown how this technique can be used to formulate nonlinear chance constraints. The method is verified by showing the ability of the Gaussian process to accurately approximate the probability density function of the underlying system and by the closed-loop behaviour of the algorithm via Monte Carlo simulations on an economic batch reactor case study.**

I. INTRODUCTION

The only advanced control method that has been employed to a significant extent in industry is model predictive control (MPC). MPC refers to a control approach that explicitly uses a dynamic model to evaluate a sequence of control actions at each sampling time by solving an optimal control problem (OCP). The success of MPC can be largely attributed to its ability to deal with multivariable plants and process constraint [1].

Many problems however are affected by uncertainties, including inaccuracies from the parameters in the dynamic model and external disturbances. Robust MPC (RMPC) methods have been proposed to handle uncertain systems for which the uncertainties are assumed to be in a bounded set [2]. For robust nonlinear MPC (RNMPC), min-max NMPC [3] and tube-based NMPC [4] have been introduced among others. These approaches enable analysis of the stability and performance of the system in the worst-case, which may however have a very small chance of occurrence and hence lead to a too conservative solution [5].

An alternative to robust MPC is given by stochastic MPC (SMPC), which assumes the uncertainties to be described by known probability density functions (pdf). Constraints in this context are given by chance or expectation constraints. SMPC alleviates the previously described problem by allowing for a level of constraint violation in probability, which leads to a trade-off between risk of constraint violation and closed-loop control performance [6].

Most work in SMPC has been on linear systems, e.g. stochastic tube based MPC [7], [8], scenario-based MPC [9], [10] and affine-parametrization approaches [11], [12], while stochastic NMPC (SNMPC) has received relatively little attention [6]. This can be in part explained by the difficulty of propagating stochastic uncertainties through a nonlinear system model without being prohibitively expensive. An exception are Markovian systems with finite possible realizations of the stochastic uncertainties, for which efficient algorithms are available [13]. Several methods have been proposed to propagate uncertainties through nonlinear systems, such as Monte Carlo sampling (MC), generalized polynomial chaos expansions (gPCe), Gaussian closure, equivalent linearization and stochastic averaging [14].

A simple procedure to solve the SNMPC problem for moderately nonlinear systems is given by successive linearization and application of linear SMPC algorithms, such as stochastic tube based MPC [15]. In [16], [17] stochastic averaging is applied using the unscented transformation, which is computationally efficient, but similar to the linearization approach only applicable to moderately nonlinear systems. [18] used a sampling average approach to obtain a tractable OCP formulation. The required number of samples was reduced by employing variance reduction techniques. In [19] Markov Chain MC was used to solve the nonlinear MPC problem, which however quickly becomes prohibitive in complexity. In particular, the Markov Chain MC approach suggested tries to find the global optimum, which is only applicable to low dimensional problems. For continuous time the Fokker-Planck equations can be used to predict the pdf of the states over time, which has been used in [20]. A Lyapunov function is included in the SNMPC formulation to guarantee probabilistic stability, however feasibility is not ensured and the method is expensive due to the requirement of solving a partial differential equation system online.

Lastly, much of the research in SNMPC has been concerned with the application of gPCe, which describes a procedure of propagating uncertainties through a nonlinear model as an efficient alternative to MC sampling by utilising orthogonal polynomials [21]. For SNMPC these methods rely on running several realizations of the uncertain parameters, while solving a least-squares problem to calculate the coefficients of the orthogonal polynomials for every iteration of the control inputs [22], [5], which is known as non-intrusive gPCe. Alternatively, Galerkin projection may be used to determine the coefficients of the orthogonal polynomial, which

1 E. Bradford is with the Faculty of Information Technology and Electrical Engineering, Engineering Cybernetics, NTNU, 7491 Trondheim, Norway eric.bradford@ntnu.no

2 L. Imsland is with the Faculty of Information Technology and Electrical Engineering, Engineering Cybernetics, NTNU, 7491 Trondheim, Norway lars.imsland@ntnu.no

however is only applicable to polynomial-type systems [23]. The chance constraints are either reformulated as second-order cone constraints or using direct MC sampling on the polynomial chaos expansion itself [23]. [24] shows how gPCe can in addition be used for additive disturbances of nonlinear systems, while [25] utilises gPCe to solve a MPC problem for model maintenance by designing experiments online. While gPCe constitute a promising approach for SNMPC, there are several disadvantages. The complexity of gPCe grows exponentially with the number of uncertain parameters, orthogonal polynomials of high orders are prone to unstable swings, time-varying disturbances are difficult to handle and lastly the expansion is only exact as the number of terms tends to infinity [26], [6].

In the statistics community the use of gPCe is rare and instead Gaussian processes (GP) are used for uncertainty quantification in "Bayesian calibration" by running different realizations of the uncertain parameters [27], [28]. An excellent comparison of gPCe to GPs is given in [26].

Gaussian process models are probabilistic, non-parametric models that not only provide predictions, but also prediction uncertainties. GPs have been shown to be a powerful tool in single- [29] and multi-objective optimization [30] by exploiting the uncertainty measure to sample functions efficiently. GPs have found various applications in MPC. GPs have been shown to be an efficient alternative to neural network models to identify nonlinear models from data for NMPC [31] with successful application to a gas-liquid separation plant [32]. In addition, GPs have been used to identify disturbance models online. For example, in [33] the GP is used to overcome unmodelled periodic error and in [34] to update a model after a fault has occurred. In [35] it is shown how the GP can be used to model residual model uncertainty and formulate chance constraints based on the Gaussian distribution of the states.

In this paper we propose the use of GPs as an alternative to non-intrusive gPCe for SNMPC. The main advantage of using GPs over gPCe in SNMPC is the fact that the uncertainty involved from the approximation of the true model by a finite number of samples is taken into account, which is otherwise ignored by the gPCe. In addition, GPs are not prone to unstable swings and are interpolating, i.e. pass exactly through all sample points provided, but otherwise suffer from the same drawbacks as gPCe. The novelty in this paper is the application of GPs to learn the mapping between uncertain parameters and model outputs for SNMPC applications.

The remainder of the paper is structured as follows. In Sec. II a stochastic nonlinear OCP is formulated. Sec. III introduces Gaussian Process regression with equations to estimate the exact mean and variance from uncertain inputs. In Sec. IV it is shown how the stochastic nonlinear OCP can be approximately solved by employing GPs and how this can be applied in a receding horizon fashion for SNMPC. Lastly, Sec. V tests the approach on a batch reactor case study by comparing open-loop predictions of the pdf and closed-loop performance via Monte Carlo simulations.

II. Stochastic Nonlinear Optimal Control Problem Formulation

In this work we consider a general discrete time stochastic nonlinear system with parametric uncertainties:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) \tag{1}$$

where k represents the discrete time, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ are the states, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ are the control inputs, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ denotes the uncertain model parameters and $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_x}$ represents the nonlinear system dynamics. The parametric uncertainties are assumed to be jointly Gaussian distributed with known mean $\mathbf{m}_{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta}$ and known covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta \times n_\theta}$, which fully specifies the pdf of $\boldsymbol{\theta}$.

Based on (1), we formulate an OCP. Assuming that the system states are measured at all times, a general OCP can be given as follows:

Finite-horizon nonlinear OCP with chance constraints

$$\begin{aligned}
&\underset{\mathbf{u}_N}{\text{minimize}} && \mathbb{E}\left(J(N, \mathbf{x}_t, \mathbf{u}_N, \boldsymbol{\theta})\right) + \omega \mathrm{Var}\left(J(N, \mathbf{x}_t, \mathbf{u}_N, \boldsymbol{\theta})\right) \\
&\text{subject to} \\
&\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) && \forall k \in \{0, ..., N-1\} \\
&\mathbb{P}(g_j^{(k)}(\mathbf{x}(k), \mathbf{u}(k), \boldsymbol{\theta}) \leq 0) \geq 1 - p_j^{(k)} \\
&&& \forall (k, j) \in \{1, \ldots, N\} \times \{1, \ldots, n_g^{(k)}\} \\
&\mathbf{u}(k) \in \mathbb{U}_k && \forall k \in \{0, ..., N-1\} \\
&\mathbf{x}(0) = \mathbf{x}_t
\end{aligned} \tag{2}$$

where the length of the time horizon is given by N, the objective consists of the expectation and variance of a nonlinear function $J(N, \mathbf{x}_t, \mathbf{u}_N, \boldsymbol{\theta})$ weighted by ω, $g_j^{(k)} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ are individual nonlinear chance constraints of which there are $n_g^{(k)}$ for each discrete time k, $p_j^{(k)} \in [0, 1] \subset \mathbb{R}$ is the desired probability of constraint violation with respect to $g_j^{(k)}$, the input constraints are represented by $\mathbb{U}_k$, $\mathbf{u}_N := \{\mathbf{u}(0), \ldots, \mathbf{u}(N-1)\}$ is a collection of inputs and lastly $\mathbf{x}_t$ is the initial state, which is assumed to be known.

The goal of the stochastic nonlinear OCP is to calculate an optimal control sequence over a finite time horizon that adjusts the pdfs of the states to obtain the optimal value of the probabilistic objective function, while allowing for a predefined violation of the stochastic nonlinear constraints.

III. Gaussian Process Regression

In this section we give a short introduction to GP regression specific for our purposes. For a more general review, please refer to [36], [37]. GP regression describes the inference of an unknown function $\xi : \mathbb{R}^{n_\theta} \to \mathbb{R}$ from data. The purpose of GPs in our case is to determine an unknown transformation $\xi(\boldsymbol{\theta})$ with respect to the uncertain parameters, hence the input dimension n_θ.

GPs describe a distribution over functions and can be seen as a generalisation of multivariate Gaussian distributions. A GP, $\xi \sim GP(m(\cdot), k(\cdot, \cdot))$, is specified by a mean function $m(\cdot)$ and a covariance function $k(\cdot, \cdot)$, defined as follows:

$$m(\boldsymbol{\theta}) := \mathbb{E}_\xi\left(\xi(\boldsymbol{\theta})\right) \tag{3}$$

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') := \mathbb{E}_\xi\left((\xi(\boldsymbol{\theta}) - m(\boldsymbol{\theta}))(\xi(\boldsymbol{\theta}') - m(\boldsymbol{\theta}'))\right) \tag{4}$$

where $\boldsymbol{\theta}$, $\boldsymbol{\theta}' \in \mathbb{R}^{n_\theta}$ are arbitrary input vectors and $\mathbb{E}_\xi(\cdot)$ is the expectation over the function space. The mean function can be interpreted as the 'average' shape of the function, while the covariance function specifies the covariance between any two function values computed at the corresponding inputs.

GPs in regression are used to place a prior on admissible functions in a Bayesian framework. The prior is given by the choice of the mean function and the covariance function. In this study we apply a constant mean function and the squared-exponential (SE) covariance function [36]:

$$m(\boldsymbol{\theta}) \coloneqq c \tag{5}$$

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \alpha^2 \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}')^T \boldsymbol{\Lambda} (\boldsymbol{\theta} - \boldsymbol{\theta}')\right) \tag{6}$$

where c is a constant, $\boldsymbol{\Lambda} = \frac{1}{\ell^2}\mathbf{I}$, ℓ is a width scaling parameter and α^2 is the signal variance. By selecting the SE covariance function we assume that the underlying transformation to be inferred is smooth and stationary.

Next we require data of the function, i.e. evaluations of the function at specific input values. We assume we are given n_s such function values at n_s different inputs. Let $\boldsymbol{\Theta} = \left[\widetilde{\boldsymbol{\theta}}_0, \widetilde{\boldsymbol{\theta}}_1, \ldots, \widetilde{\boldsymbol{\theta}}_{n_s-1}\right] \in \mathbb{R}^{n_\theta \times n_s}$ denote a data matrix with a collection of training inputs with a corresponding output vector of function evaluations defined by $\mathbf{y} = [\xi(\widetilde{\boldsymbol{\theta}}_0), \ldots, \xi(\widetilde{\boldsymbol{\theta}}_{n_s-1})]^T \in \mathbb{R}^{n_s}$.

The hyperparameters that define the prior of the GP in (5) and (6) are jointly given by the vector $\boldsymbol{\Psi} = [c, \ell, \alpha]^T$. The hyperparameters are generally unknown a priori, such that they need to be inferred from data. Maximum likelihood estimation (MLE) is commonly carried out to determine $\boldsymbol{\Psi}$. The log-likelihood of the observed data, ignoring constant terms, is given by:

$$\mathscr{L}(\boldsymbol{\Psi}) = -\frac{n_s}{2}\log(\alpha^2) - \frac{1}{2}\log(|\mathbf{K}|) - \frac{(\mathbf{y} - \mathbf{1}c)^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{1}c)}{2\alpha^2} \tag{7}$$

where $K_{ij} = \exp\left(-\frac{1}{2}(\widetilde{\boldsymbol{\theta}}_i - \widetilde{\boldsymbol{\theta}}_j)^T \boldsymbol{\Lambda} (\widetilde{\boldsymbol{\theta}}_i - \widetilde{\boldsymbol{\theta}}_j)\right)$ for all $i, j = [0, \ldots, n_s - 1]$

By setting the derivatives with respect to α^2 and c to zero, it is possible to obtain closed-form expressions for the optimal MLE values of α^2 and c as functions of $\mathbf{K}$ [38]:

$$\hat{c} = \frac{\mathbf{a}\mathbf{y}}{b} \tag{8}$$

$$\hat{\alpha}^2 = \frac{\boldsymbol{\nu}^T \mathbf{K}^{-1} \boldsymbol{\nu}}{n_s} \tag{9}$$

where $\mathbf{a} = \mathbf{1}^T\mathbf{K}^{-1}$, $b = \mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}$, $\boldsymbol{\nu} = (\mathbf{y} - \mathbf{1}\hat{c})$, $\hat{c}$ and $\hat{\alpha}^2$ are the optimal MLE values of c and α^2 respectively.

The value of the scaling parameter ℓ, was fixed in this work due to the excessive cost of evaluating it online. This will however lead to a worse fit for the GP and a correspondingly larger uncertainty of the model. An alternative could be to find a value for ℓ using (7) for different nonlinear function values $\mathbf{y}$ that gives on average the best likelihood value. Possible function values $\mathbf{y}$ can be found by open-loop simulation with different control inputs. Fixing ℓ allows for the pre-computation of various quantities. The heuristics that was used in this work for fixing ℓ can be found in [39] and is given as the median of all pairwise euclidean distances between the uncertain parameter values in the data matrix $\boldsymbol{\Theta}$:

$$\ell = \text{median}(||\widetilde{\boldsymbol{\theta}}_i - \widetilde{\boldsymbol{\theta}}_j||_2) \tag{10}$$

Finally, we require the mean and variance of $\xi(\boldsymbol{\theta})|\boldsymbol{\Theta}, \mathbf{y}$ at an arbitrary input $\boldsymbol{\theta}$, where the input follows a Gaussian distribution with mean $\mathbf{m}_{\boldsymbol{\theta}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$. $\xi(\boldsymbol{\theta})|\boldsymbol{\Theta}, \mathbf{y}$ in this case refers to the posterior function of $\xi(\cdot)$, which corresponds to the prior that was updated using the data in $\boldsymbol{\Theta}$ and $\mathbf{y}$. GPs are commonly used for deterministic inputs. The case of Gaussian distributed uncertain inputs has, however, received extensive attention for the propagation of uncertainties in the case of multi-step ahead predictions [40]. It has been shown that for our choice of mean and covariance functions given in (5) and (6) respectively, it is possible to calculate the exact mean and variance. The expressions for the expectation and variance of $\xi(\boldsymbol{\theta})|\boldsymbol{\Theta}, \mathbf{y}$ are [41]:

$$\mathbb{E}(\xi(\boldsymbol{\theta})|\boldsymbol{\Theta}, \mathbf{y}) = \hat{c} + \hat{\alpha}^2 \mathbf{d}\boldsymbol{\nu} \tag{11}$$

$$\text{Var}(\xi(\boldsymbol{\theta})|\boldsymbol{\Theta}, \mathbf{y}) = \hat{\alpha}^2 + \boldsymbol{\nu}^T \mathbf{C}\boldsymbol{\nu} - \hat{\alpha}^2 e - \left(\hat{\alpha}^2 \mathbf{d}\boldsymbol{\nu}\right)^2 \tag{12}$$

where $\mathbf{d} = \mathbf{q}^T\mathbf{K}^{-1}$, $\mathbf{C} = \mathbf{K}^{-1}\mathbf{Q}\mathbf{K}^{-1}$, $e = \text{tr}(\mathbf{K}^{-1}\mathbf{Q})$, $q_j = |\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{\Lambda} + \mathbf{I}|^{-1/2}\exp(-\frac{1}{2}(\mathbf{m}_{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_j)^T(\boldsymbol{\Sigma}_{\boldsymbol{\theta}} + \boldsymbol{\Lambda}^{-1})^{-1}(\mathbf{m}_{\boldsymbol{\theta}} - \widetilde{\boldsymbol{\theta}}_j)$ and

$$\begin{aligned} Q_{ij} = &\exp\left(-\frac{1}{2}(\widetilde{\boldsymbol{\theta}}_i - \mathbf{m}_{\boldsymbol{\theta}})^T \boldsymbol{\Lambda} (\widetilde{\boldsymbol{\theta}}_i - \mathbf{m}_{\boldsymbol{\theta}})\right) \\ &\times \exp\left(-\frac{1}{2}(\widetilde{\boldsymbol{\theta}}_j - \mathbf{m}_{\boldsymbol{\theta}})^T \boldsymbol{\Lambda} (\widetilde{\boldsymbol{\theta}}_j - \mathbf{m}_{\boldsymbol{\theta}})\right) |\mathbf{R}|^{-1/2} \\ &\times \exp\left((\mathbf{x}_\mathbf{b} - \mathbf{m}_{\boldsymbol{\theta}})^T \mathbf{R}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \boldsymbol{\Lambda} (\mathbf{x}_\mathbf{b} - \mathbf{m}_{\boldsymbol{\theta}})\right) \end{aligned}$$

where $\mathbf{R} = 2\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\boldsymbol{\Lambda} + \mathbf{I}$ and $\mathbf{x}_\mathbf{b} = \boldsymbol{\Lambda}^{-1}(\widetilde{\boldsymbol{\theta}}_i - \mathbf{m}_{\boldsymbol{\theta}}) + \boldsymbol{\Lambda}^{-1}(\widetilde{\boldsymbol{\theta}}_j - \mathbf{m}_{\boldsymbol{\theta}})$.

It should be noted that while we assume Gaussian distributed uncertainties in this paper, it is possible to determine (11) and (12) for other types of uncertainties, such as uniformly distributed uncertainties. Therefore, the method can be easily extended to other types of uncertainties.

In the GP-SNMPC algorithm the input training design $\boldsymbol{\Theta}$ is created offline and remains the same online. In addition, the hyperparameter ℓ is fixed once $\boldsymbol{\Theta}$ is fixed based on the heuristic in (10). The terms $\mathbf{a}$, b, $\mathbf{K}^{-1}$, $\mathbf{d}$, $\mathbf{C}$ and e in (8-12) are only functions of $\boldsymbol{\Theta}$ and ℓ, and hence can be pre-computed offline. This makes the use of GPs viable for SNMPC, since otherwise expensive calculations would have to be carried out for each iteration of the optimization algorithm, such as the inversion of the matrix $\mathbf{K}$ or the calculation of $\mathbf{C}$.

IV. Gaussian Process Stochastic Nonlinear Model Predictive Control

In this section we show how GPs can be exploited for reformulating the OCP given in (2). Firstly, we outline how the chance constraints in (2) can be reformulated robustly in terms of the mean and variance of the random variable in question. Subsequently, the principle of the GP-SNMPC is highlighted. Lastly, the space-filling parameter design used

is outlined and the reformulated OCP problem is given in terms of the resulting samples from the parameter design.

A. Robust chance constraints

The probabilistic control problem in (2) can be solved efficiently by robust reformulation of the chance constraints using the Chebyshev inequality, which results in the following theorem [5]:

Theorem 1: Consider a generic probability constraint of the form:

$$\mathbb{P}(\xi \leq 0) \geq 1-\varepsilon, \quad \varepsilon \in (0,1) \subset \mathbb{R} \tag{13}$$

where $\xi \in \mathbb{R}$ is some random variable with known mean $\mathbb{E}(\xi) = \hat{\xi}$ and variance $\mathrm{Var}(\xi) = \sigma_\xi^2$. Let Ω be the set of random variables with mean $\hat{\xi}$ and variance σ_ξ^2, then for any $\varepsilon \in (0,1)$, the distributionally robust probability constraint

$$\inf_{\xi \in \Omega} \mathbb{P}(\xi \leq 0) \geq 1-\varepsilon \tag{14}$$

can be shown to be equivalent to:

$$\kappa_\varepsilon \sigma_\xi + \hat{\xi} \leq 0, \quad \kappa_\varepsilon = \sqrt{(1-\varepsilon)/\varepsilon} \tag{15}$$

where σ_ξ is the standard deviation of ξ. ■

The probability constraints in the stochastic OCP in (2) are given in the form of:

$$\mathbb{P}(g_j^{(k)}(\mathbf{x}(k),\mathbf{u}(k),\boldsymbol{\theta}) \leq 0) \geq 1-p_j^{(k)} \tag{16}$$

Using Thm. 1 we arrive at the following deterministic constraints for the probability constraint in (16):

$$\kappa_{jk}\sqrt{\mathrm{Var}\left(g_j^{(k)}(\mathbf{x}(k),\mathbf{u}(k),\boldsymbol{\theta})\right)} + \mathbb{E}\left(g_j^{(k)}(\mathbf{x}(k),\mathbf{u}(k),\boldsymbol{\theta})\right) \leq 0 \tag{17}$$

where $\kappa_{jk} = \sqrt{(1-p_j^{(k)})/p_j^{(k)}}$

It should be noted that Chebyshev inequality leads to conservative constraints, in particular for pdf's that are close to Gaussian. An alternative to this has been used in gPCe based SNMPC by sampling the random variable instead to approximate the probability constraint [23].

B. GP-SNMPC Principle

Before outlining the exact equations necessary to simplify (2), we will first highlight the principle behind the approach. The main difficulty of the OCP in (2) is the determination of the statistics of the objective function $J(N,\mathbf{x}_t,\mathbf{u}_N,\boldsymbol{\theta})$ and the nonlinear functions constituting the probabilistic constraints $g_j^{(k)}(\mathbf{x}(k),\mathbf{u}(k),\boldsymbol{\theta})$. Given that $\boldsymbol{\theta}$ is assumed to be time invariant, once all control inputs $\mathbf{u}_N$ are fixed, the values of the states $\mathbf{x}(k)$ and consequently the values of the objective $J(N,\mathbf{x}_t,\mathbf{u}_N,\boldsymbol{\theta})$ and the constraints $g_j^{(k)}(\mathbf{x}(k),\mathbf{u}(k),\boldsymbol{\theta})$ depend solely on the value of $\boldsymbol{\theta}$. The functions can consequently be expressed in the following form:

$$y = \xi(\boldsymbol{\theta}) \tag{18}$$

where $\xi(\boldsymbol{\theta})$ denotes an arbitrary transformation of $\boldsymbol{\theta}$.

The problem may now be expressed as the requirement to estimate the pdf of the random variable y given the distribution of $\boldsymbol{\theta}$. In many cases it is sufficient to determine the expectation (first moment) and variance (second moment) of y. SNMPC using GPs is a scenario based approach similar to the non-intrusive polynomial chaos method proposed in [5], [22]. This involves creating several realizations of the uncertain variable $\boldsymbol{\theta}$. Each of these realizations corresponds to a separate nonlinear dynamic equation system given by (1) with $\boldsymbol{\theta}$ replaced by the respective realization.

The principle on which the procedure works is illustrated in Fig. 1. Each sample of $\boldsymbol{\theta}$ creates a separate trajectory from the initial state $x(0)$, as shown by the lines on the left-hand side graph. These trajectories each have distinct values of x at each discrete time, highlighted by the markers on the same graph. If we are now interested in the statistics of a nonlinear transformation $g(x)$, these values need to be transformed as shown by the arrows. This gives us several values for each uncertain parameter realization, which we can represent as an unknown transformation of $\boldsymbol{\theta}$, $\xi(\boldsymbol{\theta})$, as is illustrated on the right-hand side graph. GP regression is then used to estimate the unknown transformation $\xi(\boldsymbol{\theta})$. The resulting GP surrogate of $\xi(\boldsymbol{\theta})$ is then used to estimate the mean and variance from the closed-form expressions in (11) and (12). The GP needs to be rebuilt each time $\mathbf{u}_N$ changes using the data from the different scenarios, i.e. for each iteration step of the optimization algorithm.

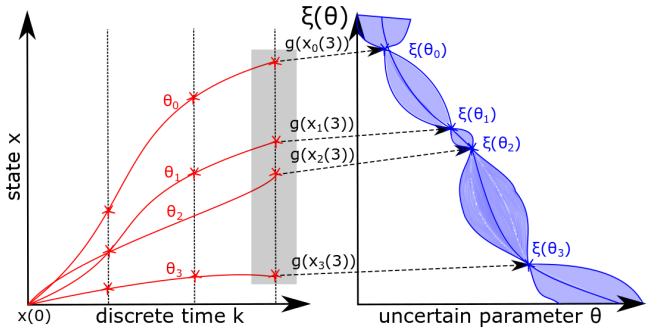


Fig. 1. Illustration of GP-SNMPC algorithm: On the left-hand side graph the trajectories are shown for each realization of $\boldsymbol{\theta}$ with markers highlighting the different values of the state x. For the final discrete time the values of the states are transformed through $g(x)$, which gives us several values, which are plotted on the right-hand side graph against the realization values of $\boldsymbol{\theta}$. It is then shown on the right-hand side graph that the unknown relationship of the transformation with $\boldsymbol{\theta}$ can be approximated by GP regression

C. Gaussian process optimal control formulation

The GP-SNMPC algorithm is a sampling-based algorithm, i.e. we create a set of different values of $\boldsymbol{\theta}$, which was represented by $\boldsymbol{\Theta} := \left[\widetilde{\boldsymbol{\theta}}_0, \widetilde{\boldsymbol{\theta}}_1, \ldots, \widetilde{\boldsymbol{\theta}}_{n_s-1}\right]$ in the previous section, where each $\widetilde{\boldsymbol{\theta}}$ implies a separate nonlinear dynamic system given by (1). From these separate simulations we then obtain the output values $\mathbf{y}$, which are used to estimate the necessary statistics in (19). The parameter design needs to ensure a good spread of $\widetilde{\boldsymbol{\theta}}$ values in the region of significant probability densities. In this work we used min-max Latin hypercube sampling from the Gaussian distribution of $\boldsymbol{\theta}$ to generate the necessary parameter set by using the procedure

described in [42]. Based on this parameter design $\boldsymbol{\Theta}$, the GP-SNMPC OCP problem may be given as follows:

Finite horizon GP-SNMPC problem with chance constraints

$$
\begin{aligned}
&\underset{\mathbf{u}_N}{\text{minimize}} \quad \mathbb{E}\left(\xi_J|\boldsymbol{\Theta},\mathbf{y}_J\right)+\omega\text{Var}\left(\xi_J|\boldsymbol{\Theta},\mathbf{y}_J\right)\\
&\text{subject to}\\
&\mathbf{x}_i(k+1)=\mathbf{f}(\mathbf{x}_i(k),\mathbf{u}(k),\widetilde{\boldsymbol{\theta}}_i)\\
&\qquad\qquad \forall(k,i)\in\{1,\ldots,N-1\}\times\{0,\ldots,n_s-1\}\\
&\kappa_{jk}\sqrt{\text{Var}\left(\xi_{g_{jk}}|\boldsymbol{\Theta},\mathbf{y}_{g_{jk}}\right)}+\mathbb{E}\left(\xi_{g_{jk}}|\boldsymbol{\Theta},\mathbf{y}_{g_{jk}}\right)\leq 0\\
&\kappa_{jk}=\sqrt{(1-p_j^{(k)})/p_j^{(k)}}\quad \forall(k,j)\in\{1,\ldots,N\}\times\{1,\ldots,n_g^{(k)}\}\\
&\hat{c}_J=\frac{\mathbf{a}\mathbf{y}_J}{b},\qquad \hat{\alpha}_J^2=\frac{\boldsymbol{\nu}_J^T\mathbf{K}^{-1}\boldsymbol{\nu}_J}{n_s}\\
&\hat{c}_{g_{jk}}=\frac{\mathbf{a}\mathbf{y}_{g_{jk}}}{b},\quad \hat{\alpha}_{g_{jk}}^2=\frac{\boldsymbol{\nu}_{g_{jk}}^T\mathbf{K}^{-1}\boldsymbol{\nu}_{g_{jk}}}{n_s}\\
&\mathbb{E}\left(\xi_J|\boldsymbol{\Theta},\mathbf{y}_J\right)=\hat{c}_J+\hat{\alpha}_J^2\mathbf{d}\boldsymbol{\nu}_J\\
&\text{Var}\left(\xi_J|\boldsymbol{\Theta},\mathbf{y}_J\right)=\hat{\alpha}_J^2+\boldsymbol{\nu}_J^T\mathbf{C}\boldsymbol{\nu}_J-\hat{\alpha}_J^2e-\left(\hat{\alpha}_J^2\mathbf{d}\boldsymbol{\nu}_J\right)^2\\
&\mathbb{E}\left(\xi_{g_{jk}}|\boldsymbol{\Theta},\mathbf{y}_{g_{jk}}\right)=\hat{c}_{g_{jk}}+\hat{\alpha}_{g_{jk}}^2\mathbf{d}\boldsymbol{\nu}_{g_{jk}}\\
&\text{Var}\left(\xi_{g_{jk}}|\boldsymbol{\Theta},\mathbf{y}_{g_{jk}}\right)=\hat{\alpha}_{g_{jk}}^2+\boldsymbol{\nu}_{g_{jk}}^T\mathbf{C}\boldsymbol{\nu}_{g_{jk}}-\hat{\alpha}_{g_{jk}}^2e-\left(\hat{\alpha}_{g_{jk}}^2\mathbf{d}\boldsymbol{\nu}_{g_{jk}}\right)^2\\
&\mathbf{u}(k)\in\mathbb{U}_k\qquad \forall k\in\{0,...,N-1\}\\
&\mathbf{x}_i(0)=\mathbf{x}_t\qquad \forall i\in\{0,\ldots,n_s-1\}
\end{aligned}
\tag{19}
$$

where $\mathbf{x}_i$ corresponds to the state vector of scenario i with uncertain parameter $\widetilde{\boldsymbol{\theta}}_i$, as mentioned in Sec. III the terms $\mathbf{a}$, b, $\mathbf{K}^{-1}$, $\mathbf{d}$, $\mathbf{C}$ and e can be pre-computed offline from $\boldsymbol{\Theta}$, $\mathbf{y}_J=[J(N,\mathbf{x}_t,\mathbf{u}_N,\widetilde{\boldsymbol{\theta}}_i),\ldots,J(N,\mathbf{x}_t,\mathbf{u}_N,\widetilde{\boldsymbol{\theta}}_i)]^T$ is a vector of values of the objective function for each scenario, $\mathbf{y}_{g_{jk}}=[g_j(\mathbf{x}_0(k),\mathbf{u}(k),\widetilde{\boldsymbol{\theta}}_0),\ldots,g_j(\mathbf{x}_{n_s-1}(k),\mathbf{u}(k),\widetilde{\boldsymbol{\theta}}_{n_s-1})]^T$ is a vector of values for each nonlinear chance constraint for each scenario, $\boldsymbol{\nu}_{g_{jk}}=\mathbf{y}_{g_{jk}}-\mathbf{1}\hat{c}_{g_{jk}}$ and $\boldsymbol{\nu}_J=\mathbf{y}_J-\mathbf{1}\hat{c}_J$

Algorithm 1: GP-SNMPC

Initialize:

- Supply uncertain parameter description: $\boldsymbol{m}_\theta$ and $\boldsymbol{\Sigma}$
- Create uncertain parameter design $\boldsymbol{\Theta}$
- Calculate $\mathbf{a}$, b, $\mathbf{K}^{-1}$, $\mathbf{d}$, $\mathbf{C}$ and e from (8-12)
- Define OCP in (19)

At each sampling time $t=0,1,2,\ldots$

- Take measurement $\mathbf{x}_t$
- Solve (19) to obtain $\mathbf{u}_N$
- Apply the first control input from $\mathbf{u}_N$, $\mathbf{u}(0)$ to the real system

A few remarks regarding the computational complexity of the algorithm. Firstly, the expressions involving the expectation and variance are either linear or quadratic with respect to the scenario output, which yields an overall well-posed optimization problem that is smooth everywhere and makes implementation relatively easy. The algorithm is the most effective when there are a small number of constraints compared to the number of states, since then only a small number of GPs are required, the computational cost of which are likely negligible.

V. Stochastic Nonlinear Model predictive control of a batch reactor

A. Dynamic model equations and OCP formulation

In this section the algorithm is verified on a semi batch reactor with a cooling jacket. The reaction is the saponification of ethyl acetate, which is a good example of an exothermic reaction for which safety concerns are paramount to prevent a thermal runaway. The dynamic model was taken from [43]:

$$\dot{C}_A=-k(C_AC_B-C_CC_D/K_C)-FC_A/V, \tag{20a}$$
$$\dot{C}_B=-k(C_AC_B-C_CC_D/K_C)+F(\exp(\theta_2)-C_B)/V, \tag{20b}$$
$$\dot{C}_C=k(C_AC_B-C_CC_D/K_C)-FC_C/V, \tag{20c}$$
$$\dot{N}_W=FC_{W0}, \tag{20d}$$
$$\dot{T}=\frac{\dot{Q}_{gs}-\dot{Q}_{rs}}{NC_p}, \tag{20e}$$
$$\dot{V}=F \tag{20f}$$

where $k=0.0039175\exp(5472.7(1/273-1/T))$, $K_C=10^{\exp(\theta_3)/T}$, $C_D=C_C$, $C_w=N_W/V$, $\dot{Q}_{gs}=Vk(C_AC_B-C_CC_D/K_C)\Delta H_{RX}$, $\dot{Q}_{rs}=F\exp(\theta_1)(\exp(\theta_2)+C_{W0})(T-T_0)+UA(T-T_j)$, $NCP=V\left(C_{P_A}C_A+\exp(\theta_1)(C_B+C_C+C_D+C_W)\right)$,$C_A,C_B,C_C,C_D$ are the concentration of species A, B, C and D respectively in kmol.m^{-3}, T is the reactor temperature in K, N_W the amount of water in the reactor in kmol and V the reactor volume in m^3.

Three parameters were overall assumed to be uncertain indicated by components of $\boldsymbol{\theta}$, which are assumed to be Gaussian distributed with mean and covariance given by:

$$\mathbf{m}_{\boldsymbol{\theta}}=\begin{bmatrix}4.40\\0.59\\8.26\end{bmatrix},\boldsymbol{\Sigma}_{\boldsymbol{\theta}}=\begin{bmatrix}2\cdot10^{-3}&5\cdot10^{-6}&-2\cdot10^{-4}\\5\cdot10^{-6}&2.5\cdot10^{-3}&-2\cdot10^{-4}\\-2\cdot10^{-4}&-2\cdot10^{-4}&1\cdot10^{-2}\end{bmatrix} \tag{21}$$

For the missing parameters refer to [43], example 13-3.

The control input is the feedrate to the semi-batch reactor given by F. In compact form we can write $\mathbf{x}=[C_A,C_B,C_C,N_W,T,V]^T$ and $u=F$. Using discretization the equations can be given as a discrete time equation system with the discredization time set to 15s. Direct orthogonal collocation was used for the discretization of the dynamic equation system in (20) with 4th order polynomials placed according to the Radau quadrature rule.

$$\mathbf{x}(k+1)=\mathbf{f}(\mathbf{x}(k),u(k),\boldsymbol{\theta}) \tag{22}$$

Based on this equation system we formulate an OCP as

follows:

$$
\begin{aligned}
&\underset{\mathbf{u}_N}{\text{minimize}} && -\mathbb{E}\left(C_C(N)V(N)\right)+1.5\text{Var}\left(C_C(N)V(N)\right)\\
&\text{subject to}\\
&\mathbf{x}(k+1)=\mathbf{f}(\mathbf{x}(k),u(k),\boldsymbol{\theta}) && \forall k\in\{0,...,N-1\}\\
&\mathbb{P}(T_{adiabatic}(k)-320\leq 0)\geq 0.95 && \forall k\in\{1,\ldots,N\}\\
&u(k)\in[0,8\times 10^{-3}] && \forall k\in\{0,...,N-1\}\\
&\mathbf{x}(0)=\mathbf{x}_t
\end{aligned}
\tag{23}
$$

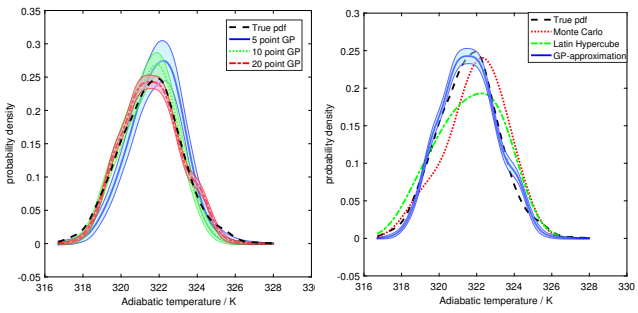
where $T_{adiabatic}(k)$ refers to the adiabatic temperature change at discrete time k defined as $T_{adiabatic}(k)=T(k)-\Delta H_{RX}C_B(k)V(k)/NCP(k)$, assuming that C_A is in excess. The adiabatic temperature change refers to the maximum attainable temperature in the reactor under a cooling failure.

The OCP in (23) aims to maximize the mean production of C with trade-off towards the variation of the production of C, while keeping the adiabatic temperature below 320K in probability to prevent a thermal runaway. The stochastic OCP in (23) is converted using GPs. Algorithm 1 shows how this OCP can then be used for a receding-horizon implementation to obtain a stochastic MPC. The time horizon was set to $N=15$ with a final batch time of $240s$. The OCP was solved utilising Casadi [44]. The resulting nonlinear programming problem was solved by employing Ipopt [45]. Lastly, IDAS [46] was applied to simulate the "real" nonlinear equation system. At time $t=0$, the the initial state to solve the OCP was set to $\mathbf{x}_0=[25,1,0,6.6,0.2,310]$.

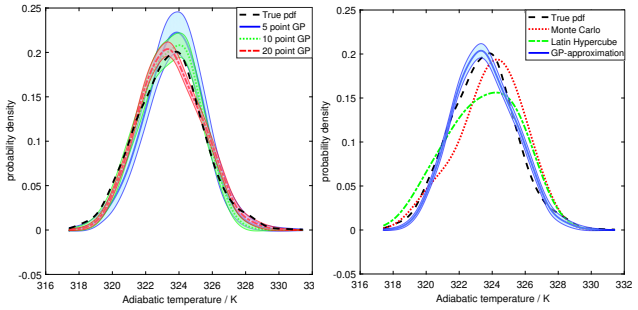
B. Open-loop predictions of statistics

We first evaluate the ability of the GPs to estimate the statistics of the relevant random variables. The procedure was taken from [47]. In particular, the random variables "adiabatic temperature" at discrete times $k=5$ and $k=15$ and "amount of species C" at the end of the batch were chosen for the analysis, since these represent the nonlinear probabilistic constraint and the objective respectively. To carry out the comparison the feed rate was fixed at $F=8\times 10^{-3}\text{m}^3/\text{s}$ and the uncertain parameter distribution was sampled 1000 times to obtain the "true" pdf. This is contrasted to the approximations of the pdfs obtained from the GP-approximations built from 5-points, 10-points and 20-points, which is shown in the graphs in Fig. 2 on the left-hand side. The pdfs were obtained from kernel density estimation. In addition, it is possible for GPs to calculate a confidence region for the pdfs, since a GP corresponds to not just one function value, but a distribution of possible function values. We can see that for all 3 pdfs the approximations get closer to the "true" pdf as the number of points increases. In addition, it can be seen that the confidence region is largest for the GP with 5 points and smallest for the GP with 20 points, which reflects the observations made. For 20 points the GP more or less matches the pdf in all 3 cases.

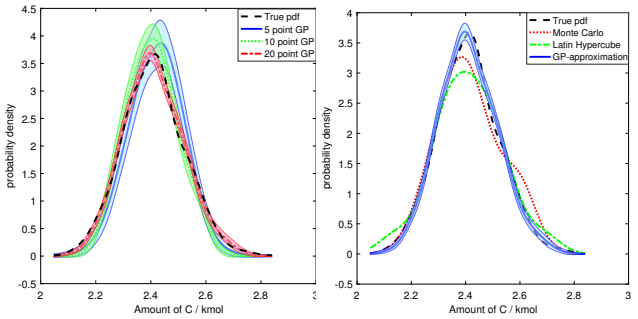
Furthermore, the 20-point GP approximations were compared on the right-hand side graphs to the pdf estimates obtained from 20 Monte Carlo samples and the 20 "maximin" Latin Hypercube samples the GP was built from. We can see that the GP approximation gives the most accurate representation of all pdfs, while the "maximin" Latin Hypercube tends to overestimate the variance and the Monte Carlo tends to underestimate the variance.



(a) Adiabatic temperature at $k=5$



(b) Adiabatic temperature at $k=15$



(c) Amount of C at $k=15$

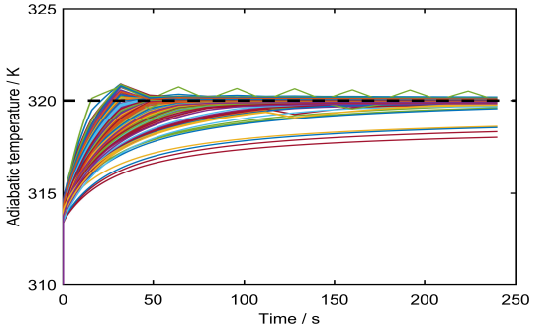
Fig. 2. pdf estimation comparison results for the OCP in (23) with fixed $F=8\times 10^{-3}\text{m}^3/\text{s}$. Left graph: true pdf compared to GP approximations with 5 points, 10 points and 20 points. For the GP plots a 99% confidence region is shown. Right graph: true pdf compared to Monte Carlo, Latin hypercube and GP-approximation built on a sample size of 20 points.

C. Closed-loop implementation

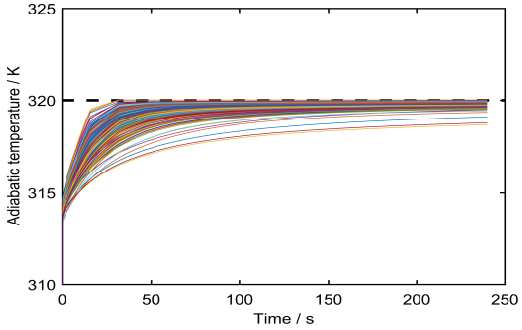
Next, the OCP in (23) was solved in a shrinking horizon fashion to gauge the closed-loop performance of the algorithm with the size of the uncertain parameter design set to 20. To verify the robustness of the approach 200 Monte Carlo simulations were performed by sampling 200 independent realizations of the uncertain parameter from the Gaussian multivariate distribution. These uncertain parameters are used for the simulation of the "real" system to which the control inputs of the GP-SNMPC algorithm were applied. The algorithm is compared to a nominal NMPC, again using a shrinking horizon implementation. The OCP of the nominal NMPC implementation is similar to the OCP in (23), except that the predictions are taken to be deterministic by fixing

the uncertain variables to their nominal values. Therefore, the objective is given by the predicted amount of C at the final time. To avoid infeasibilities the adiabatic temperature constraint was formulated using soft constraints.

In Fig. 3 the trajectories of the adiabatic temperature constraint are shown. It can be seen that the nominal NMPC algorithm violates the upper bound significantly for many of the 200 simulations. Overall, 51% of the simulations for the NMPC algorithm lead to constraint violations. It can be seen that the violations mostly occur at the beginning of the reaction due to the initial concentration of B being high. Once the concentration of B becomes lower towards the later parts of the reaction, the probability of constraint violation is lower, since then the adiabatic temperature change corresponds more or less to the actual temperature of the reactor with less uncertainty. The GP-SNMPC on the other hand fulfilled the constraint in all simulations.



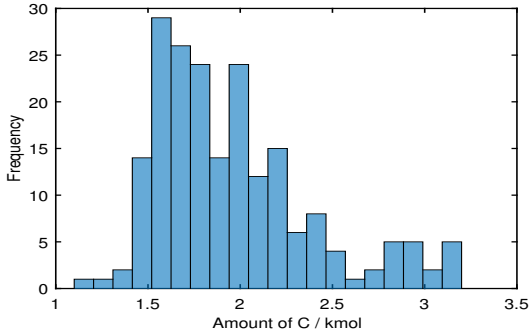
(a) Nominal nonlinear model predictive control



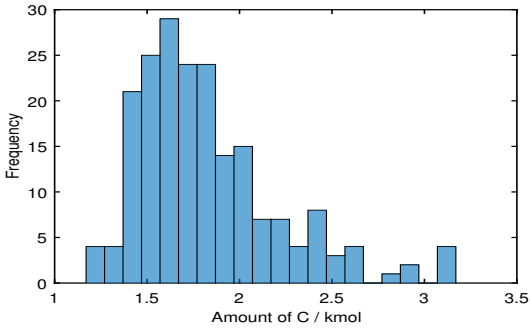
(b) Stochastic nonlinear model predictive control

Fig. 3. 200 Monte Carlo trajectories of the adiabatic temperature constraint. The constraint at 320K is highlighted by a dashed black line.

In Fig. 4 the histograms of the amount of C produced at the end of the batch are shown based on the 200 Monte Carlo simulations for both methods. The objective of the GP-SNMPC was given in terms of both the mean and variance, while the nominal NMPC was geared towards maximizing the amount of C only. The GP-SNMPC produced on average 1.83 kmol of C, while the nominal NMPC algorithm produced 1.96 kmol on average, which is approximately 7% more. On the other hand the variance of the GP-SNMPC is 25% less, hence the performance differences may be due to a different control objective or due to the conservativeness of the GP-SNMPC algorithm.



(a) Nominal nonlinear model predictive control



(b) Stochastic nonlinear model predictive control

Fig. 4. Histograms of the amount of C at the end of the batch based on the 200 Monte Carlo simulations

VI. Conclusions

In conclusion, a novel framework for SNMPC has been proposed by employing GPs to handle Gaussian distributed parametric uncertainties with known mean and covariance. The SNMPC problem involved both objective and probability constraints based on general nonlinear functions. The probability constraints were reformulated requiring only mean and variance. GP-SNMPC is a scenario-based MPC algorithm that uses the data from the various realizations of the uncertain parameter to build a surrogate, from which the variance and mean of the nonlinear objective and constraint functions are estimated efficiently. In addition, GP-SNMPC not only takes into account the variance induced by the uncertain parameters but also, unlike gPCe, the uncertainty of the surrogate itself. A semi-batch reactor case study showed that GP-SNMPC is able to provide predictions on the statistics of the problem more accurately than either Monte Carlo or Latin hypercube samples. Lastly, a shrinking horizon application showed excellent closed-loop performance by ensuring the fulfilment of a nonlinear chance constraint, while optimizing a probabilistic objective.

VII. Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675215. Eric Bradford would like to thank BASF SE for providing an industrial placement.

References

[1] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[2] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 1999, pp. 207–226.

[3] H. Chen, C. W. Scherer, and F. Allgower, "A game theoretic approach to nonlinear robust receding horizon control of constrained systems," in *Proceedings of the 1997 American Control Conference*, vol. 5. IEEE, 1997, pp. 3073–3077.

[4] D. Q. Mayne, E. C. Kerrigan, E. J. Van Wyk, and P. Falugi, "Tubebased robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[5] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *2014 American Control Conference*. IEEE, 2014, pp. 2413–2419.

[6] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems*, vol. 36, no. 6, pp. 30–44, 2016.

[7] M. Cannon, B. Kouvaritakis, and D. Ng, "Probabilistic tubes in linear stochastic model predictive control," *Systems & Control Letters*, vol. 58, no. 10, pp. 747–753, 2009.

[8] M. Cannon, Q. Cheng, B. Kouvaritakis, and S. V. Raković, "Stochastic tube MPC with state estimation," *Automatica*, vol. 48, no. 3, pp. 536–541, 2012.

[9] M. Prandini, S. Garatti, and J. Lygeros, "A randomized approach to stochastic model predictive control," in *IEEE 51st Annual Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 7315–7320.

[10] D. Bernardini and A. Bemporad, "Scenario-based model predictive control of stochastic constrained linear systems," in *Proceedings of the 48th IEEE Conference on Decision and Control*. IEEE, 2009, pp. 6333–6338.

[11] F. Oldewurtel, C. N. Jones, and M. Morari, "A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback," in *47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4731–4736.

[12] P. Hokayem, E. Cinquemani, D. Chatterjee, F. Ramponi, and J. Lygeros, "Stochastic receding horizon control with output feedback and bounded controls," *Automatica*, vol. 48, no. 1, pp. 77–88, 2012.

[13] P. Patrinos, P. Sopasakis, H. Sarimveis, and A. Bemporad, "Stochastic model predictive control for constrained discrete-time Markovian switching systems," *Automatica*, vol. 50, no. 10, pp. 2504–2514, 2014.

[14] U. Konda, P. Singla, T. Singh, and P. D. Scott, "State uncertainty propagation in the presence of parametric uncertainty and additive white noise," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 5, p. 51009, 2011.

[15] M. Cannon, D. Ng, and B. Kouvaritakis, "Successive linearization NMPC for a class of stochastic nonlinear systems," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 249–262.

[16] E. Bradford and L. Imsland, "Stochastic Nonlinear Model Predictive Control with State Estimation by Incorporation of the Unscented Kalman Filter," *arXiv preprint arXiv:1709.01201*, 2017.

[17] E. Bradford and L. Imsland, "Economic Stochastic Model Predictive Control Using the Square Root Unscented Kalman Filter," in *IFAC Advanced Control of Chemical Processes*, 2018, p. in press.

[18] E. Bradford and L. Imsland, "Expectation constrained stochastic nonlinear model predictive control of a batch bioreactor," in *27 European Symposium on Computer Aided Process Engineering*. Elsevier, 2017, vol. 40, pp. 1621–1626.

[19] A. L. Visintini, W. Glover, J. Lygeros, and J. Maciejowski, "Monte Carlo optimization for conflict resolution in air traffic control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 470–482, 2006.

[20] E. A. Buehler, J. A. Paulson, and A. Mesbah, "Lyapunov-based stochastic nonlinear model predictive control: Shaping the state probability distribution functions," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 5389–5394.

[21] Z. K. Nagy and R. D. Braatz, "Distributional uncertainty analysis using power series and polynomial chaos expansions," *Journal of Process Control*, vol. 17, no. 3, pp. 229–240, 2007.

[22] L. Fagiano and M. Khammash, "Nonlinear stochastic model predictive control via regularized polynomial chaos expansions," in *51st Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 142–147.

[23] S. Streif, M. Karl, and A. Mesbah, "Stochastic nonlinear model predictive control with efficient sample approximation of chance constraints," *arXiv preprint arXiv:1410.4535*, 2014.

[24] V. A. Bavdekar and A. Mesbah, "Stochastic nonlinear model predictive control with joint chance constraints," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 270–275, 2016.

[25] V. A. Bavdekar and A. Mesbah, "Stochastic model predictive control with integrated experiment design for nonlinear systems," *IFAC-PapersOnLine*, vol. 49, no. 7, pp. 49–54, 2016.

[26] A. O'Hagan, "Polynomial Chaos: A tutorial and critique from a statistician's perspective," 2013.

[27] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.

[28] A. O'Hagan, "Bayesian analysis of computer code outputs: a tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 10, pp. 1290–1300, 2006.

[29] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[30] E. Bradford, A. M. Schweidtmann, and A. Lapkin, "Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm," *Journal of Global Optimization*, pp. 1–33, 2018.

[31] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 3. IEEE, 2004, pp. 2214–2219.

[32] B. Likar and J. Kocijan, "Predictive control of a gasliquid separation plant based on a Gaussian process model," *Computers & chemical engineering*, vol. 31, no. 3, pp. 142–152, 2007.

[33] E. D. Klenske, M. N. Zeilinger, B. Schölkopf, and P. Hennig, "Gaussian process-based predictive control for periodic error correction," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 110–121, 2016.

[34] J. M. Maciejowski and X. Yang, "Fault tolerant control using Gaussian processes and model predictive control," in *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*. IEEE, 2013, pp. 1–12.

[35] L. Hewing and M. N. Zeilinger, "Cautious Model Predictive Control using Gaussian Process Regression," *arXiv preprint arXiv:1705.10702*, 2017.

[36] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in neural information processing systems*, 1996, pp. 514–520.

[37] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[38] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[39] T. S. Jaakkola, M. Diekhans, and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies." in *ISMB*, vol. 99, 1999, pp. 149–158.

[40] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting," *Advances in neural information processing systems*, pp. 545–552, 2003.

[41] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, 2011, pp. 465–472.

[42] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[43] H. S. Fogler, "Elements of chemical reaction engineering," 1999.

[44] J. Andersson, "A general-purpose software framework for dynamic optimization," *Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10*, 2013.

[45] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[46] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[47] N. E. Owen, P. Challenor, P. P. Menon, and S. Bennani, "Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 5, no. 1, pp. 403–435, 2017.