Journal of Process Control 24 (2014) 358–367

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Probabilistic analysis and control of systems with uncertain parameters over non-hypercube domain

Pham Luu Trung Duong, Moonyong Lee*

School of Chemical Engineering, Yeungnam University, Gyeongsan, Republic of Korea

ARTICLE INFO

Article history:
Received 30 June 2013
Received in revised form 5 January 2014
Accepted 11 February 2014

Keywords:
Polynomial chaos
Robust controller design
Stochastic uncertainty
Non-hypercube domain
Fractional order system

ABSTRACT

Generalized polynomial chaos expansion provides a computationally efficient way of quantifying the influence of stochastic parametric uncertainty on the states and outputs of a system. In this study, a polynomial chaos-based method was proposed for an analysis and design of control systems with parametric uncertainty over a non-hypercube support domain. In the proposed method, the polynomial chaos for the hypercube domain was extended to non-hypercube domains through proper parameterization to transform the non-hypercube domains to hypercube domains. Based on the proposed polynomial chaos framework, a constrained optimization problem minimizing the mean under the maximum allowable variance was formulated for a robust controller design of dynamic systems with the parametric uncertainties of the non-hypercube domain. Several numerical examples ranging from integer to fractional order systems were considered to validate the proposed method. The proposed method provided superior control performance by avoiding the over-bounds from a hypercube assumption in a computationally efficient manner. From the simulation examples, the computation time by gPC analysis was approximately 10–100 times lower than the traditional approach.

1. Introduction

Modeling inaccuracies or uncertainties are inevitable in practice. The deterministic worst case setting is the most popular way of considering parametric uncertainty in control theory [1–3]. One of the drawbacks of this worst case-based method is its computational complexity, in that it often becomes computational intractable for general uncertainty structures [4]. To avoid this problem, the uncertainties are normally assumed to be of an interval or hypercube type in the worst case approaches. On the other hand, when the uncertainties are not of the hypercube type, they need to be over bounded by a hypercube, which can lead to an excessively conservative design. Therefore, a probabilistic approach that can handle the actual type of uncertainties should be considered for a proper controller design.

The Monte-Carlo (MC) method is a representative traditional probabilistic approach for the analysis and control of uncertain systems [4–6]. The brute-force implementation of the MC method involves first generating an ensemble of random realizations with each parameter drawn randomly from its uncertainty distribution. Solvers are then applied to each member to obtain an ensemble of results. The ensemble of results is then post-processed to estimate the relevant statistical properties, such as the mean, standard deviation, and density function. The stability and robustness of the system against the uncertainties can be inferred from these statistical properties. The estimation of the mean converges with the inverse square root of the number of runs, which makes MC methods computationally expensive. The high computational cost of MC methods has motivated the development of computationally efficient methods for uncertainty propagation and quantification that replaces or accelerates them, such as Quasi Monte Carlo (QMC) methods [4,7,8] and generalized polynomial chaos methods (gPC) [9–12].

Many studies have considered probabilistic approach for robust controller design [4,13–15]. These studies, however, focused primarily on MC/QMC methods rather than the gPC method. The most popular approach is to consider linear systems with stochastic additive Gaussian input [13]. For a linear time invariant system, the states are also Gaussian. Therefore, the predictive control problem is formulated as a standard chance constraint problem. When parametric uncertainties are involved, as reported in reference [14], the scenario approach is suggested for robust model predictive control. In reference [15], the MC method was used to design a full state feedback controller in mini unmanned aerial vehicles. Please refer to [16] for a recent update of the probabilistic methods for control system design.

* Corresponding author. Tel.: +82 053 810 2512.
E-mail address: mynlee@yu.ac.kr (M. Lee).

http://dx.doi.org/10.1016/j.jprocont.2014.02.012

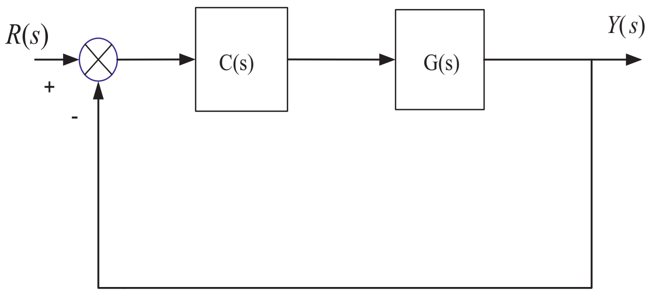

Fig. 1. Closed-loop control system.

This study considered generalized polynomial chaos (gPC) expansions as a functional surrogate model of a system model in the presence of uncertainty of the non-hypercube type. Although uncertainty propagation and quantification using gPC expansions have been studied extensively, the application of gPC to probabilistic robust control is relatively new [17,18], and unavailable to systems with uncertainty of the non-hypercube domain.

In this study, a polynomial chaos-based method was proposed for probabilistic analysis and robust controller design for systems with parametric uncertainty over the non-hypercube support domain.

This paper is organized as follows. Section 2 introduces the theory of gPC methods for uncertainty with a hypercube domain. Section 3 derives several parameterizations for an analysis of a non-hypercube domain. In Section 4, the gPC based method for the fraction order controller design is proposed. Section 5 includes several examples of probabilistic analysis and controller design of the systems with non-hypercube uncertainty.

2. Probabilistic analysis using polynomial chaos theory for hypercube domain

Consider a closed loop control system in Fig. 1 with a plant and a controller:

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu};$$
$$G(s) = \frac{b_m s^{\beta_m} + \ldots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + \ldots + a_0 s^{\alpha_0}} e^{-Ls} \tag{1}$$

where the vector of its parameter $\xi = (a_1, \ldots, a_n, b_1, \ldots, b_m, L) = (\xi_1, \xi_2, \ldots, \xi_N)$ is a random vector of mutually independent uniform random components with probability density functions of $\rho_i(\xi_i) : \Gamma_i \to \mathbb{R}^+$. Therefore, the joint probability density of the random vector, ξ, is $\boldsymbol{\rho} = \prod_{i=1}^{N} \rho_i$, and the support of ξ is $\boldsymbol{\Gamma} \equiv \prod_{i=1}^{N} \Gamma_i \in \mathbb{R}^N$. The set of one-dimensional orthonormal polynomials, $\{\phi_i(\xi_i)_{m=0}^{d_i}\}$, can be defined in finite dimension space, Γ_i, with respect to the weight, $\rho_i(\xi)$. Based on a one-dimensional set of polynomials, an N-variate orthonormal set can be constructed with P total degrees in the space, $\boldsymbol{\Gamma}$, using the tensor product of the one-dimensional polynomials that satisfies:

$$\int_{\Gamma} \Phi_m(\xi)\Phi_n(\xi)\rho(\xi)d\xi = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases} \tag{2}$$

Considering a response function of the system output $f(y(t, \xi))$ with the statistics (e.g. mean and variance) of interest, the N-variate Pth order approximation of the response function can be constructed as follows:

$$f_N{}^P(y(t,\xi)) = \sum_{i=1}^{M} \widehat{f}_m(t)\Phi_m(\xi);$$
$$M + 1 = \begin{pmatrix} N+P \\ N \end{pmatrix} = \frac{(N+P)!}{N!P!} \tag{3}$$

where P is the order of polynomial chaos, and $\widehat{f}_m$ is the coefficient of the gPC expansion that satisfies (2) as follows:

$$\widehat{f}_m = \mathbf{E}[\Phi_m f(y)] = \int_{\Gamma} f(y)\Phi_m(\xi)\rho(\xi)d\xi \tag{4}$$

where $\mathbf{E}[]$ denotes the expectation operator.

The probabilistic collocation approach [9] was used to obtain the gPC coefficients of the response function because of its simplicity. The algorithm is expressed briefly as follows:

- Choose a collocation set, $\{\xi_i^{(m)}, w^{(m)}\}_{m=1}^{qi}$ for each random component, ξ_i, for every direction $i = 1, \ldots, N$, and construct a one-dimensional integration rule,

$$Q_{q_i}^{(i)}[g] = \sum_{j=1}^{q_i} g(\xi_i^{(j)}) w_i^{(j)} \tag{5}$$

where Q[ ] denotes the quadrature approximation of the univariate integration. A Gaussian quadrature [11] is normally used as a one-dimensional integral rule in classical spectral methods, such as the deterministic equivalent modeling method (DEMM).

- Obtain an N-dimensional integration rule by the tensorization of the one-dimensional integral rule:

$$\ell^{Q}[g] = (Q_{q_1}{}^{(1)} \otimes \ldots \otimes Q_{q_N}{}^{(N)})[g]$$
$$= \sum_{j_1=1}^{q_1} \ldots \sum_{j_N=1}^{q_N} g(\xi_1^{(j_1)}, \ldots, \xi_N^{(j_N)})(w_1{}^{(j_1)} \otimes \ldots \otimes w_N{}^{(j_N)}) \simeq \int_{\Gamma} g(\xi)\rho(\xi)d\xi \tag{6}$$

where $\otimes$ and $\ell^{Q}[]$ denote the tensor product and the multivariate quadrature (cubature) approximation, respectively.

- Approximate the gPC coefficients in (4) using the numerical integration rule in (6).

$$\widehat{f}_j = \ell^{Q}[f(y,\xi)\Phi(\xi)\boldsymbol{\rho}] = \sum_{m=1}^{Q} f(\xi^{(m)})\Phi_j(\xi^{(m)})w^{(m)}\, for\, j = 1, \ldots, M \tag{7}$$

where $\widehat{f}$ is the numerical approximation of $\widehat{f}$ using cubature.

- Construct an N-variate Pth order gPC approximation of the response function in the form, $\tilde{f}_N^P = \sum_{j=1}^{M} \widehat{f}_j \Phi_j(\xi)$.

Once all the gPC coefficients have been evaluated, a post-processing procedure is then carried out to obtain the statistics of the response function, $f(y(t, \xi))$.

The mean of the response function is the first expansion coefficient,

$$\mathbf{E}[\tilde{f}_N^P] = \mu_f = \int_{\Gamma} \tilde{f}_N^P \rho(\xi)\, d\xi = \int_{\Gamma} \left[\sum_{j=1}^{M} \widehat{f}_j(\xi)\Phi_j(\xi) \right] \rho(\xi)\, d\xi = \widehat{f}_1 \tag{8}$$

The variance of the response function $f(y(t, \xi))$ can be expressed as

$$D_f = \sigma_f^2 = \mathbf{E}[(f-\mu_f)^2] = \int_{\Gamma}\left(\sum_{j=1}^{M}\widehat{f}_j(\xi)\Phi_j(\xi)-\widehat{f}_1\right)\left(\sum_{j=1}^{M}\widehat{f}_j(\xi)\Phi_j(\xi)-\widehat{f}_1\right)\rho(\xi)\,d\xi = \sum_{j=2}^{M}\widehat{f}_j^2 \tag{9}$$

Eqs. (8) and (9) employ the property that the polynomial set begins with $\Phi_1(\xi) = 1$. The weight function of the polynomial is the probability density function. If a response function, $f(y) = y$, is chosen, the mean and variance of the system's states can be approximated using Eqs. (8) and (9), respectively. The surrogate gPC series, $\tilde{f}_N^P = \sum_{j=1}^{M}\widehat{f}_j\Phi_j(\xi)$, can be sampled to obtain the probability density function for the response function.

The set, $\{\phi_i\}_{i=1}^{d_i}$, is the orthonormal polynomial of ξ_i with a weight function, $\rho_i(\xi_i)$. The weight function is the probability density function of a random variable, ξ_i. This links the distribution of the random variable, ξ_i, and the type of the orthogonal polynomials in its gPC-based method. Because the uncertainties are a uniform type, the Legendre polynomials and their quadrature should be used for optimal convergence. More details on how to construct the Legendre polynomials and its associated quadrature can be found in references [9,10] and the references therein.

The errors for stochastic simulation have two sources: truncation error and simulation error. Truncation error is due to the finite dimensional approximation of the infinite dimension random process (finite order of gPC expansion, finite number of samples). Simulation error is due to the error of the deterministic method used for the simulated system with the sample/cubature set.

For information on the spectral convergence of the gPC approach under the smooth dependence of the solution on the random parameters, please refer to [9–12].

Remark. The computational effort needed by the gPC method is scaled with the number of cubature nodes used to estimate the gPC coefficients. Therefore, the computational load of gPC increases exponentially with increasing dimensions of random space. For this reason, the gPC method is suggested only when the number of random parameters is low, e.g. <5 [9].

3. Parameterization of non-hypercube domain

The presence of uncertainty in the system description has always been a critical issue in control theory and its applications. The most popular way when considering the uncertainties is to consider it as an interval uncertainty (i.e. with a hypercube domain). Nevertheless, the non-hypercube probability spaces with an irregular shape of the parameter domain often occur in control engineering problems, e.g. as reported in [19–22] and the references therein. According to references [21–23], the parameter ellipsoidal uncertainty descriptions are obtained more naturally from parameter identification procedures. When the uncertainties belong to the non-hypercube domain, the uncertainty set is normally over bounded with the hypercube domain to simplify the design procedure, which often leads to a too conservative design.

According to reference [4], the uniform distribution may have the worst-case properties compared to other distributions. Therefore, the uniform distribution is considered in the plan as a distribution-free robustness design. In this study, the design and analysis with the parameters distributed uniformly in a non-hypercube domain were considered.

In this section, the gPC method was extended to a non-hypercube domain. First, the polynomial chaos for a hypercube domain was constructed, as in the previous section. To utilize the gPC framework for hypercube domains, proper parameterizations to transform a non-hypercube domain to a hypercube domain are presented for several popular types of non-hypercube domains: ellipsoidal, simplex, etc.

The purpose of considering the simplex uncertainty is that polytopic and other complex uncertainty domains can be decomposed easily into multiple simplex domains. In that case, under certain conditions, the multi-element polynomial chaos can be used for probabilistic analysis [24].

The following theorem was used to parameterize a non-hypercube domain in to a hypercube one [4].

Theorem 1. *Let $\xi \in \mathbb{R}^n$ be a random vector with density, $\boldsymbol{\rho}_\xi(\xi_1, \ldots, \xi_n)$, continuous in a support domain $\boldsymbol{\Gamma} \subseteq \mathbb{R}^n$, and let $\boldsymbol{\eta} = \tau(\xi)$, where $\tau : \boldsymbol{\Gamma} \to \Upsilon, \Upsilon \subseteq \mathbb{R}^n$, is a one-to-one mapping, so that the inverse $\xi = \tau^{-1}(\boldsymbol{\eta}) = h(\boldsymbol{\eta})$ is well defined. The partial derivatives, $\partial\xi_i/\partial\eta_l = \partial h_i/\partial\eta_l$, exist and are continuous on Υ.*

The random vector $\boldsymbol{\eta}$ has a density of

$$\rho_\eta(\eta) = \rho_\xi(h(\eta))J(\xi \to \eta) \tag{10}$$

where

$$J(\xi \to \boldsymbol{\eta}) = \begin{vmatrix} \dfrac{\partial\xi_1}{\partial\eta_1} & \dfrac{\partial\xi_2}{\partial\eta_1} & \cdots\cdots & \dfrac{\partial\xi_N}{\partial\eta_1} \\ \dfrac{\partial\xi_1}{\partial\eta_2} & \dfrac{\partial\xi_2}{\partial\eta_2} & \cdots\cdots & \dfrac{\partial\xi_N}{\partial\eta_2} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial\xi_1}{\partial\eta_N} & \dfrac{\partial\xi_2}{\partial\eta_N} & \cdots\cdots & \dfrac{\partial\xi_N}{\partial\eta_N} \end{vmatrix} \tag{11}$$

is the absolute value of the determinant of the transformation Jacobian.

The gPC expansion becomes non-polynomial expansions due to the transform [9]. Therefore, this transform leads to sub-optimal convergence in the gPC expansion. Note that the transformations ensure that the new random vector has an exact density function as the given one.

3.1. Parameterization of a l_2 ball and weighted l_2 ball (uniform in an ellipsoid)

3.1.1. Euclidean ball and its parameterization

A random vector, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$, is distributed uniformly in $\mathcal{B}_r$, a l_2 ball with a radius r, if the density can be expressed as follows [4]:

$$\boldsymbol{\rho}_{\boldsymbol{\eta}} = \begin{cases} \dfrac{1}{Vol(\mathcal{B}_r)} & if \quad \|\boldsymbol{\eta}\|_2 = \left(\displaystyle\sum_{i=1}^{N}|\eta_i|^2\right)^2 \le r \\ 0 & otherwise \end{cases} \tag{12}$$

where *Vol(.)* denotes the volume and $\|.\|_2$ is the l_2 norm of the vector.

3.1.2. Parameterization algorithm

- Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots\ldots\xi_N)$ be a vector of the independent uniform random components in $[-1, 1]^N$. The following transformation returns a real random vector, $\boldsymbol{\eta}$, distributed uniformly in $\mathcal{B}_r$, a l_2 ball with radius ***r:***

$$\begin{aligned} \theta_1 &= \sin\left(\frac{\pi}{2}\xi_1\right) \\ \theta_2 &= \cos\left(\frac{\pi}{2}\xi_1\right)\sin\left(\frac{\pi}{2}\xi_2\right) \\ &\dots \\ \theta_{N-1} &= \cos\left(\frac{\pi}{2}\xi_1\right)\cos\left(\frac{\pi}{2}\xi_2\right)\dots\sin\left(\frac{\pi}{2}\xi_{N-1}\right) \\ \theta_N &= \pm\cos\left(\frac{\pi}{2}\xi_1\right)\cos\left(\frac{\pi}{2}\xi_2\right)\dots\cos\left(\frac{\pi}{2}\xi_{N-1}\right) \end{aligned} \tag{13}$$

and

$$\boldsymbol{\eta} = r\theta\left(\frac{1}{2}\xi_N + \frac{1}{2}\right)^{1/N} \tag{14}$$

Consider a simple example taken from [25] to demonstrate the parameterization method, where it is important to integrate

$$\int_{x^2+y^2\le 1} [x^2 + y^2 - 2xy]\, dxdy \tag{15}$$

This integral can be rewritten as an expectation of a function with two random variables, x and y, distributed uniformly in a unit disk:

$$I = \pi\frac{1}{\pi}\int_{x^2+y^2\le 1} [x^2 + y^2 - 2xy]\, dxdy = \pi E(x^2 + y^2 - 2xy) \tag{16}$$

The following gives a list for the use of the gPC method with the above parameterization for estimating Eq. (16).

- Set the radius of uncertainty radius $r = 1$;
- Obtain the two 1D Legendre quadrature sets and their associated polynomials. The two quadratures $\{{\xi_1}^{(i)}, {w_1}^{(i)}\}_1^q$ and $\{{\xi_2}^{(j)}, {w_2}^{(j)}\}_1^q$ have the same support [−1,1].
- Using the transformation in Eqs. (13) and (14) for obtaining two new 1D quadrature rules for a disk

$$\begin{aligned} \xi_1^{(i)} &\to \nu_1^{(i)} = \left[\frac{1}{2}\xi_1^{(i)} + \frac{1}{2}\right]^{1/2} \\ \xi_2^{(j)} &\to \nu_2^{(j)} = \pi\xi_2^{(j)} \end{aligned}$$

- The two random variables x and y can be parameterized as

$$\begin{aligned} x^{(i)}(\xi_1, \xi_2) &= r{\nu_1}^{(i)}\sin({\nu_2}^{(j)}) \\ y^{(i)}(\xi_1, \xi_2) &= r{\nu_1}^{(i)}\cos({\nu_2}^{(j)}) \end{aligned}$$

- The 2D cubature and polynomial set are obtained by the tensorization of 1D sets.
- The coefficients of gPC expansions are obtained using Eq. (7)

$$\hat{f}_j = \sum_{m=1}^{Q} f(\xi^{(m)})\Phi_j(\xi^{(m)})w^{(m)} \quad \textit{for} \quad j = 1, \dots, M$$

where M is defined as Eq. (3), $Q = q^2$

- The first coefficient is the mean of the function, $f(x, y) = x^2 + y^2 - 2xy$. The integral (16) is then obtained by multiplying the first coefficients with π.

Note that only 10^2 simulations (number of cubature nodes) were sufficient to obtain an accurate result $I = 2.3563$. The Gauss Legendre quadrature was calculated using the OPQ suite [26]. On the other hand, the MC method with the code given in [25] required 500,000 simulations for the same accuracy.

3.1.3. Ellipsoid and its parameterization

A random vector, $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$, is distributed uniformly in $\varepsilon(\boldsymbol{\eta}_0, \Theta)$, which is a weighted l_2 ball with a positive definite shaping matrix, Θ, and center $\boldsymbol{\eta}_0$, if the density function can be expressed as

$$\boldsymbol{\rho}_{\boldsymbol{\eta}} = \begin{cases} \dfrac{1}{Vol(\varepsilon(\boldsymbol{\eta}_0, \Theta))} & \textit{if } (\boldsymbol{\eta} - \boldsymbol{\eta}_0)^T\Theta^{-1}(\boldsymbol{\eta} - \boldsymbol{\eta}_0) \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Parameterization algorithm:

- Compute a matrix P such that $\Theta = P^T P$.
- Assume that there is a vector $\boldsymbol{\kappa} \in \mathbb{R}^n$ distributed uniformly in a unit ball, $\mathcal{B}$
- A transformation, $\boldsymbol{\eta} = P\boldsymbol{\kappa} + \boldsymbol{\eta}_0$, will result in a vector, $\boldsymbol{\eta}$, distributed uniformly in the ellipsoid $\varepsilon(\boldsymbol{\eta}_0, \Theta)$

3.2. Parameterization of a simplex

A simplex is a generalization of the notion of a triangle in 3 or more dimensions. A simplex in N dimensions is defined by $N+1$ vertices $\zeta_0, \dots, \zeta_N$.

Suppose that $N+1$ points $\zeta_0, \dots, \zeta_N$ are independent. A simplex $Si(\zeta_0, \dots, \zeta_N)$ is a set of vectors in $\mathbb{R}^N$ of the form

$$\zeta_0 + C\nu \tag{18}$$

where C is an invertible matrix with columns $\zeta_1 - \zeta_0, \dots, \zeta_N - \zeta_0$ and $\nu = \left\{(\nu_1, \dots, \nu_n) \in \mathbb{R}^N : \sum_{i=1}^{N}\nu_i \le 1\right\}$.

Parameterization algorithm [27]:

- Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_N)$ be a vector of the independent uniform random components in $[-1, 1]^N$.
- The following transformation will result in a vector, $\boldsymbol{\eta}$, distributed uniformly in a simplex $Si(\zeta_0, \dots, \zeta_N)$:

$$\begin{aligned} \eta_1 &= (N\nu_1)^{1/N}((N-1)\nu_2)^{1/(N-1)}\dots(2\nu_{N-1})^{1/2}\nu_N \\ &\dots \\ \eta_j &= (N\nu_1)^{1/N}\prod_{i=1}^{N-j}((N-i)\nu_{i+1})^{1/(N-i)}\left(1-(j-1)\nu_{N-j+2}\right)^{1/(j-1)}, \quad j = 2, \dots, N-1; \\ \eta_N &= (N\nu_1)^{1/N}\left(1-((N-1)\nu_2)^{1/(N-1)}\right) \end{aligned} \tag{19}$$

where$\nu_j = \frac{1}{N-j+1}\left(\frac{1}{2}\xi_i + \frac{1}{2}\right)$ $j = 1, \dots, N$.

Remark. Despite there being several ways of transforming a hypercube domain to a non-hypercube domain, not all of them are suitable for the gPC method.

4. Risk analysis and controller design

Apart from the controller design, the problem of determining the probabilities that a certain system characteristic exceeds critical thresholds has significant importance. The problem with determining the probabilities that a certain system characteristic exceeds a critical value is known as risk analysis.

4.1. Risk analysis

Some notation and definitions from [4] are introduced for risk analysis of a system.

Denote the set of uncertainty as $\mathcal{B}_\xi$

In deterministic robustness, the main objective is to check if a given system property is satisfied for all possible values of the uncertainty. The satisfaction of a system property is given by the performance function, $f(\xi)$, and a performance level, γ, so that the inequality

$$f(\xi) < \gamma \tag{20}$$

guarantees all possible cases of uncertainties $\xi \in \mathcal{B}_\xi$

Denote the volume of the set of uncertainty by $vol(\mathcal{B}_\xi)$

For probabilistic analysis two sets are introduced: good and bad sets. These are a subset of $\mathcal{B}_\xi$. They are constructed so that their union is equivalent to the set of uncertainties and their intersection is empty. The good set is defined as $\mathcal{B}_G = \{\xi \in \mathcal{B}_\xi : f(\xi) \le \gamma\}$. The bad set is defined as $\mathcal{B}_B = \{\xi \in \mathcal{B}_\xi : f(\xi) > \gamma\}$.

In a probabilistic setting, the measure of robustness is given as the relative volume of the good set $\mathcal{B}_G = \{\xi \in \mathcal{B}_\xi : f(\xi) \le \gamma\}$. In other words, the volume of the good set needs to be sufficiently large, i.e. the ratio

$$\frac{vol(\mathcal{B}_G)}{vol(\mathcal{B}_\xi)} \tag{21}$$

is close to one.

The ratio given in Eq. (21) is called a performance degradation function. This actually is the probability $Pr\{f(\xi) \le \gamma\}$, which can be found by sampling the surrogate gPC model in Eq. (3) or directly from the MC/QMC simulation.

4.2. Controller design

Because a fractional-order system can be considered as a generalization for classical integer-order systems [28–31], this section provides a gPC based method for the fractional order controller design with non-hypercube uncertainties.

Consider the closed loop control system in Fig. 1, where $C(s)$ is a $PI^\lambda D^\mu$ controller [29],

$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \tag{22}$$

and $G(s)$ is a process with stochastic parametric uncertainty

$$G(s) = \frac{b_m s^{\beta_m} + \ldots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + \ldots + a_0 s^{\alpha_0}} e^{-Ls} \tag{23}$$

where a_i, b_j, L are random variables with given distributions with a non-hypercube domain.

4.2.1. Mean–variance optimization

The $\mathrm{PI}^\lambda\mathrm{D}^\mu$ controller's parameters are obtained by optimizing the cost function,

$$\min_{Kp,Ki,Kd,\lambda,\mu} J = \min_{Kp,Ki,Kd,\lambda,\mu} \int_0^T \left| M_e(t) \right| dt \tag{24}$$

subject to a constraint on the variance of the output

$$\max_{0 \le t \le T} D_y(t) \le D_{\max} \tag{25}$$

where M and D denote the mean and variance of the signal, respectively.

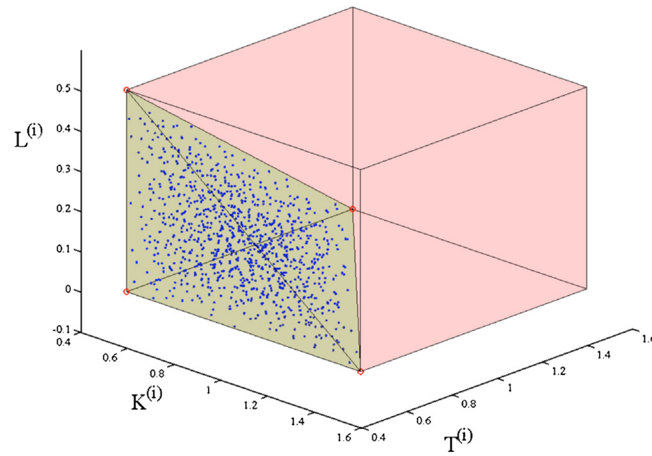

Fig. 2. Generation of 2000 uniform samples in the simplex for case 1. For the design with a cube uncertainty, a cube was used to bound the simplex.

A large variance of the system output indicates a large deviation from the nominal response under parametric uncertainties because the variance is a measure of the variability of a random process, which can lead to either very sluggish or oscillatory responses. The upper bound, $D_{\max}$, of the variance of the system output can be considered the maximum allowable sensitivity of the system response to the parametric uncertainties. Therefore, adjusting $D_{\max}$ allows trade-off between the robustness of the response to uncertainty and the response speed. The mean error can be viewed as the weighted averaged multi-model error.

Remark. The performance objective function in the frequency domain can also be used.

4.2.2. Design with probabilistically guaranteed robustness

The goal of this approach is to ensure a given probability p of the stability of a control system. For this purpose, the constraint (25) is replaced with $Pr\{\max(z_i) < 0\} \ge p$, where z_i are closed loop poles.

5. Examples

The performance of the proposed method for probabilistic analysis and controller design was evaluated using case studies.

5.1. Example 1: probabilistic analysis of FOPDT system

In this example, a first order plus dead-time (FOPDT) system, which is one of most representative and popular models in process control [32,33], was considered with a PID controller designed based on a two dimensional cube uncertainty [17],

$$G(s) = \frac{K}{Ts+1} e^{-Ls}, \quad C0(s) = 0.989 + \frac{1.104}{s} + 0.007s \tag{26}$$

Assume that K, T and L are distributed uniformly inside a simplex defined using the following four vertices:

$$\zeta_0 = [0.5; 0.5; 0]; \quad \zeta_1 = [0.5; 1.5; 0]; \quad \zeta_2 = [1.5; 0.5; 0];$$
$$\zeta_3 = [0.5; 0.5; 0.5] \tag{27}$$

A parameterization algorithm, as reported in Section 3, was applied to transform the simplex into a hypercube, $[\xi_1, \xi_2, \xi_3] \in [-1, 1]^3$. Fig. 2 shows 2000 samples of a real 3 dimensional vector of K, L, T distributed uniformly inside the simplex. The proposed approach was compared with the QMC method for predicting the output mean and variance for a step reference input (Fig. 3). An analytical representation of the output was

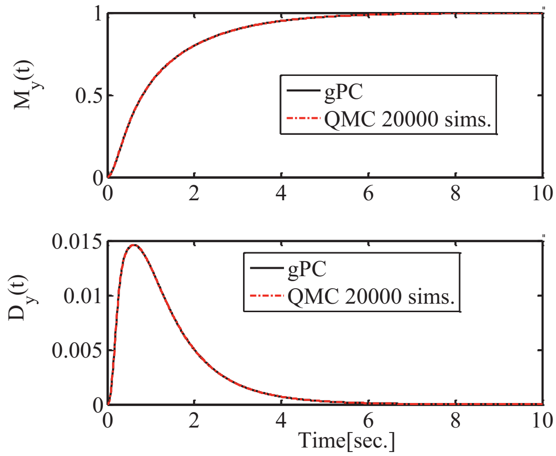

Fig. 3. Statistical characteristics (mean and variance) in Example 1.

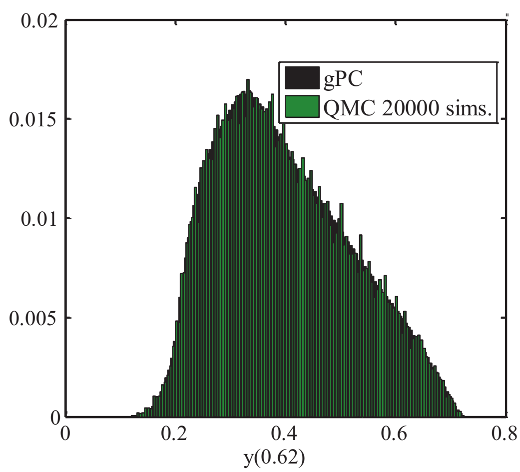

Fig. 4. Density function of the output for Example 1 at 0.62 s.

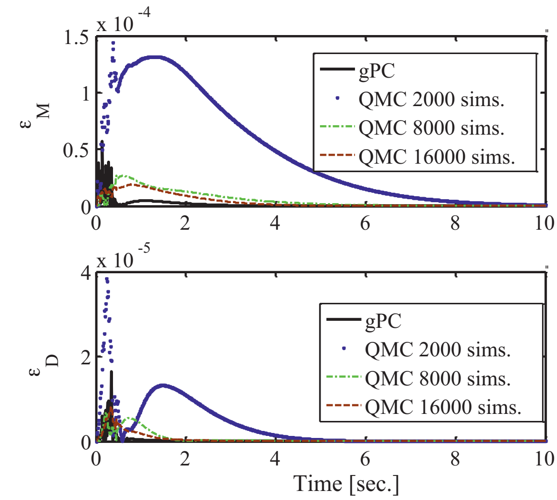

Fig. 5. Absolute errors of the means and variances for Example 1.

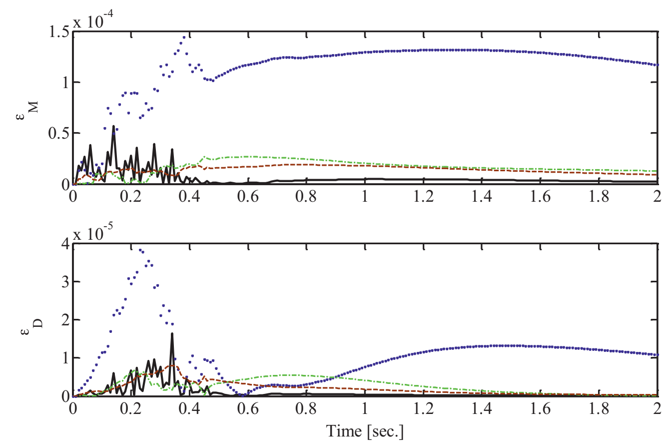

Fig. 6. Absolute errors of the means and variances (Fig. 5) with zoom into the interval [0, 2] s.

then obtained effectively in terms of $\xi = (\xi_1, \xi_2, \xi_3)$, as $\tilde{y}_N^P(t) = \sum_{j=1}^{M} \tilde{y}_j(t)\Phi_j(\xi)$. The probability density function of the output can be obtained by simple sampling of this analytical representation. Note that in the QMC method, it is important to simulate the dynamic given in Eq. (26) one thousand times at the sampling points to obtain the required probability density, whereas in the gPC method, only a small number of cubature nodes are needed to obtain the surrogate representation. For a system with a smooth response, exponential convergence is expected for the gPC method. This fast convergence property of the gPC is also reported in reference [9] and the references therein. Fig. 4 compares the density function of the output at 0.62 s, where the peak variance occurs, obtained using the gPC and QMC methods. The gPC surrogate model is a time dependent model, and can be sampled at any time moment for the density functions of the system output. The performance of different methods was compared for case 1 in terms of the absolute errors of the means and variances as follows:

$$\begin{aligned} \varepsilon_M &= \left| M_Y - {M_Y}^{20,000} \right| \\ \varepsilon_D &= \left| D_Y - {D_Y}^{20,000} \right| \end{aligned} \tag{28}$$

where $M_Y^{20,000}$ and $D_Y^{20,000}$ denote the mean and variance of the reference solution of the Monte-Carlo method with 20,000 simulations, respectively. In the gPC method, the number of curvature nodes used to estimate the gPC coefficients plays the same role as the computational complexity (number of simulation) in the QMC method. Figs. 5 and 6 show the absolute error of the QMC and proposed methods.

From Fig. 5, the degradation in the performance of gPC can be observed at the beginning of the transient, where the time delay creates discontinuity in the system output. On the other hand, the response quickly becomes smooth, and the overall accuracy recovers within a short time. Fig. 5 shows that the QMC method with a lack of samples also suffers from the same problem. The gPC method provides the best results when peak variance occurs. Despite the degradation due to a discontinuity in the beginning, the error of the gPC method in the beginning is still acceptable.

Table 1 lists the computation times needed to obtain the statistical characteristics and simulation parameters using both methods. The proposed gPC based method is much more computationally efficient than the QMC method, whereas the generated probabilistic characteristics from both methods are similar. This shows that the gPC can provide similar accuracy with far less computational effort than the QMC method.

Suppose that this simplex type of uncertainty is changed to a hypercube uncertainty (or an interval type of uncertainty), as shown in Fig. 2. Fig. 7 shows the mean and variance. If the actual uncertainty is a hypercube, the variance of the system output will increase significantly, as expected.

Table 1
Simulation parameters and time profiles for obtaining the probabilistic characteristics of the system for the gPC and the Quasi Monte-Carlo (QMC) method in Examples 1, 2 and 3.

Example	Simulation parameters		Computation time	
	QMC (Halton sampling)	gPC	QMC (Halton sampling)	gPC
1	20,000 samples	275 cubature points	1426.45 s	6.13 s
2	10,000 samples	100 cubature points	812.9 s	4.77 s
3	20,000 × 26 samples	26 × 100 cubature points	28.71	3.26 s

Remark. The controller C0 is a somewhat lofty design (it is not designed by considering these actual uncertainties) and it is only used for probabilistic performance analysis.

5.2. Example 2: probabilistic analysis of fractional order system

In this example, the H_2 performance of an uncertain fractional order system was studied. The following fractional order was considered with the closed loop configuration in Fig. 1.

$$G(s) = \frac{b}{as^{1.1}+1}; \quad C(s) = 1 \tag{29}$$

where $\boldsymbol{\eta} = [a, b]$ is a random vector distributed uniformly inside the ellipsoid, $\varepsilon(\boldsymbol{\eta}_0, \Theta)$, where $\boldsymbol{\eta}_0 = [0.85\ 2.6]^T$

$$\Theta = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix} \tag{30}$$

This model can be found in the Cole–Cole model [34–36], which provides a parameter of the fractional order, α, which improves the fit between dielectric theory and dispersion data [36]. A parameterization algorithm, as shown in Section 3, was applied to transform this ellipsoid into a hypercube $[\xi_1, \xi_2] \in [-1, 1]^2$. Fig. 8 shows 2000 samples of a real 2 dimensional vector of a and b distributed uniformly inside this ellipse. Fig. 8 also shows a rectangle used to contain this ellipse as a hypercube domain. The gPC surrogate model of a closed loop system norm was constructed using 100 cubature nodes. This effective analytical gPC model was sampled with 10,000 samples without a simulation effort or computational burden. For the QMC method, 10,000 samples were first generated and distributed uniformly in the ellipse. The H_2 norms of the sampled closed loop system were evaluated and post-processed to obtain the density. The computational burden of the QMC method comes from the repetition of calculating the H_2 norms. The H_2 norm was calculated using the CRONE toolbox [37]. Fig. 9 shows the density functions of the closed loop system norm obtained using the QMC and proposed method. Table 1 lists the computation times needed to obtain the statistical characteristics and simulation parameters using both methods.

For comparison purposes, Fig. 10 shows the density function of the closed loop system norm with the rectangle uncertainty. When using the rectangle uncertainty, the H_2 norm of the closed loop system increased by approximately 10% due to conservative design. From this example, the computational load for the probabilistic analysis of fractional order system was excessive. Therefore, the proposed method becomes more attractive in the probabilistic analysis and design of a fractional order system with a moderate number of uncertainties.

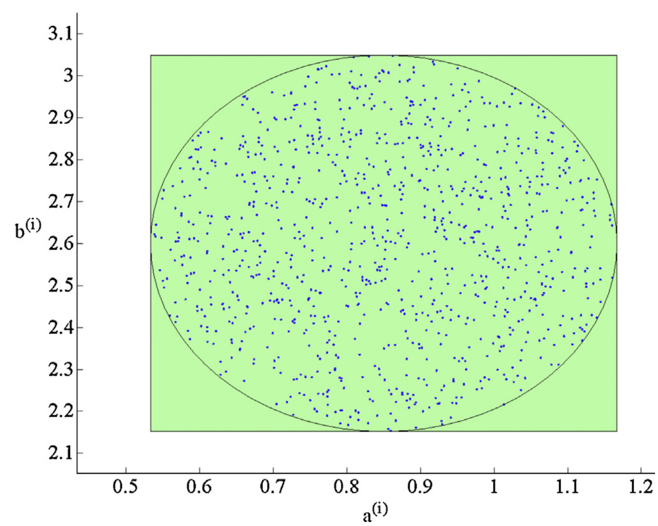

Fig. 8. Generation of 2000 uniform samples in the ellipse defined by Eq. (24).

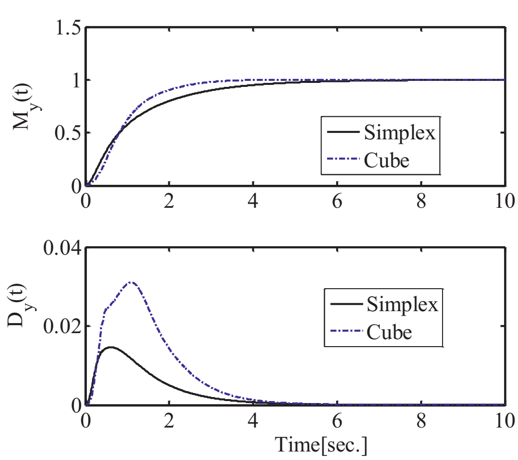

Fig. 7. Comparison of the statistical characteristics (means and variances) in Example 1 for a cube uncertainty and simplex uncertainty.

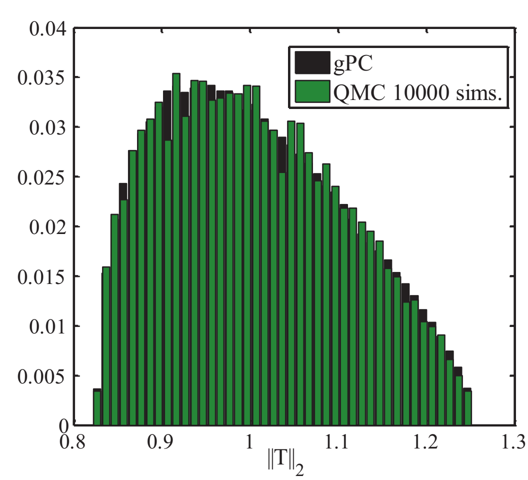

Fig. 9. Density function of the closed loop system H_2 norm for Example 2.

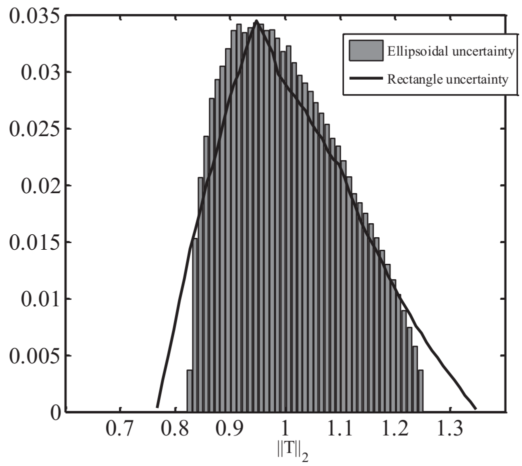

Fig. 10. Density functions of the closed loop system H_2 norm for Example 2 with a rectangle over-bounding compared to an original ellipse uncertainty.

5.3. Example 3: probabilistic robustness of uncertain systems

This example consider the stability of a 4th order integer order system with a closed-loop characteristic equation of

$$p(s,q)=2+q_1+q_2+(5+q_1+3q_2)s+(6+3q_2)s^2+(4+q_2)s^3+s^4 \tag{31}$$

where $q=(q_1,q_2)$ is distributed uniformly inside a rectangle with radius ρ

$$\mathscr{B}_q=\left\{q\in R^2:\|q\|_\infty\le\rho\right\} \tag{32}$$

Here, the performance function is the maximum real part of the zeros of the characteristic Eq. (31).

In this case, the uncertainty is uniform in a hypercube. Therefore, the Legendre polynomial chaos can be used directly to obtain the performance degradation function.

The volume of good set can be computed analytically as a function of the uncertainty radius [4]:

$$Vol(\mathscr{B}_G(\rho))=\begin{cases}4\rho^2 & if\quad \rho\le 1\\ \frac{2}{3}\rho^3-3\rho^2-(\alpha(\rho)-13)\rho-\alpha(\rho)-\frac{16}{3} & if\quad 1\le\rho\le 1.5\\ \frac{3}{2}\rho^2+4\rho+\frac{7}{24}-\frac{9}{2}\alpha(\rho)^3 & if\ 1.5\le\rho\le 3\end{cases} \tag{33}$$

where $\alpha(\rho)=1/3\sqrt{(2\rho+2)}$. The performance degradation function can then be obtained as:

$$degrad(\rho)=\frac{Vol(\mathscr{B}_G(\rho))}{Vol(\mathscr{B}_q(\rho))} \tag{34}$$

This function is depicted graphically in Fig. 11.

The performance degradation curve for a range of uncertainty radii [0.5, 3] is constructed by choosing a set of equally space grid points, $0.5=\rho_1<\rho_2<\ldots<\rho_{26}=3$, and for every grid point, performing 20,000 independent identically distributed Quasi Monte Carlo simulations. In other words, Equation (31) needs to be solved $26\times 20{,}000$ times.

The performance degradation was calculated by the proposed method with Legendre chaos of order 10. In other words, Eq. (31) only needs to be solved 100×26 times, and the 26 surrogate models are sampled with 20,000 samples. Fig. 11 and Table 1 show that the error by the gPC method is similar to the QMC method with significantly less computational time.

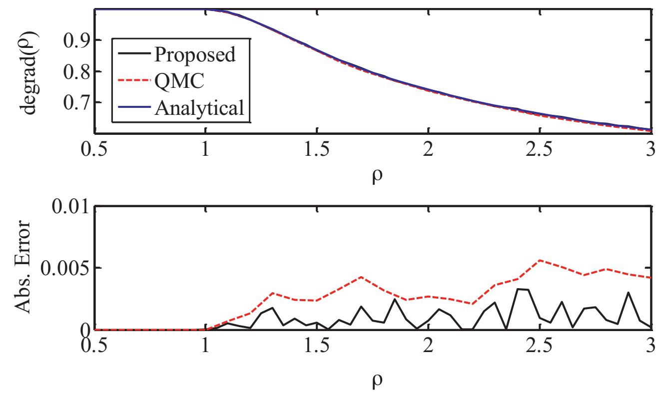

Fig. 11. Performance degradation function for Example 3 with rectangle uncertainty and its absolute error using the proposed and QMC method.

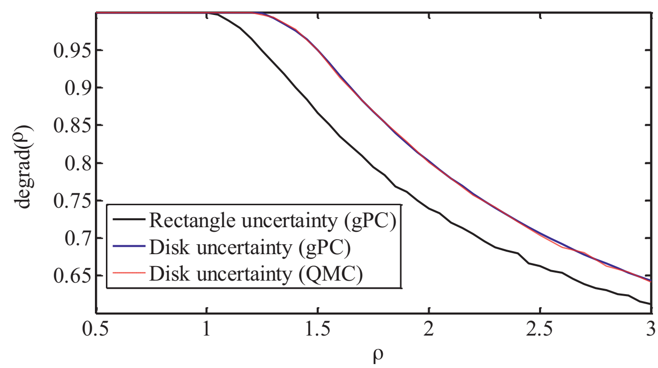

Fig. 12. Performance degradation function for Example 3 with Disk (Ellipsoid) uncertainty.

The deterministic robustness margin is $r=1$. By fixing the probability of stability to 0.95, the probabilistic stability radius is $r_{0.95}=1.25$.

Consider the uncertainty distributed uniformly in the circle

$$\mathscr{B}_q{}^D=\left\{q\in R^2:\|q\|_2\le\rho\right\} \tag{35}$$

The gPC method is used to construct the performance degradation function with respect to this uncertainty set in a simlar manner to the rectangle case. The performance degradation function is depicted in Fig. 12. For comparison, the degradation function with respect to the rectangle uncertainty is also shown in the Figure. The result by the QMC method is also shown for validation purpose.

By fixing the probability of stability to 0.95, the probabilistic stability radius in the case is now $r_{0.95}{}^D=1.45$, as expected, which is larger than the stability radius for the uncertainty of a rectangle.

5.4. Example 4: probabilistic robust controller design

From previous examples, the gPC method was more suitable for controller design by numerical optimization owing to its low computational load. In this example, a robust $PI^\lambda D^\mu$ controller was designed for a FOPDT system. Assume that the uncertainties can be modeled using a simplex in example 1. The controller parameters can be obtained by minimizing the objective functions as follows:

$$\min_{Kp,Ki,Kd,\lambda,\mu} J=\min_{Kp,Ki,Kd,\lambda,\mu}\int_0^T\left|M_e(t)\right|dt \tag{36}$$

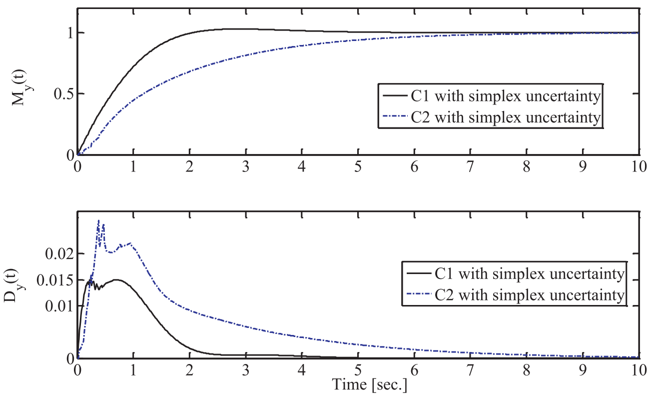

Fig. 13. Statistical characteristics (mean and variance) for a simplex uncertainty in Example 3: C1 is designed based on the simplex uncertainty assumption, whereas C2 based on a cube uncertainty assumption.

subject to a constraint on the variance of the output

$$\max_{0\le t\le T} D_y(t) \le 0.015 \tag{37}$$

The resulting parameters of the controller were C_1: $K_p = 1.153; K_i = 2.12; K_d = 0.084; \lambda = 1; \mu = 1.142$. Fig. 13 shows the mean and variance of the system with this controller. The algorithm in reference [17] was applied when a cube was used to over bound the simplex, as shown in Fig. 2. The results of the controller were C_2: $K_p = 0.624; K_i = 0.618; K_d = 0.15; \lambda = 1; \mu = 0.8241$. Fig. 13 also shows the mean and variance of the system with the controller C_2 and simplex uncertainty.

As expected, the controller design when the cube uncertainty was used was much more sluggish than that based on the simplex uncertainty. For comparison, Fig. 14 shows the bounded regions obtained from 2000 possible responses, when the system was controlled by C_1 and C_2 with the simplex uncertainties. Fig. 14 shows that the controller based on the cube or interval uncertainty provides sluggish responses due to the conservative design.

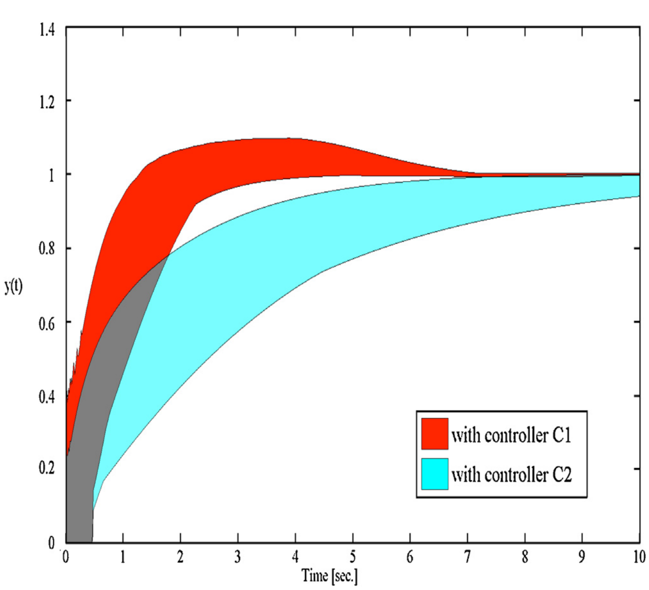

Fig. 14. Bounded region obtained from 2000 possible responses of the uncertain FOPDT systems with simplex uncertainty.

6. Conclusions

A gPC based method was proposed to examine the probabilistic behavior of systems subjected to random parametric uncertainties distributed uniformly on a non-hypercube domain. This was achieved by proper parameterization of the gPC framework from the hypercube domain. The numerical results of the well-known Quasi Monte Carlo model were used to compare the accuracy and computational efficiency of the proposed method. The proposed method was more suitable for the controller design than the QMC method. With different assumptions of the distribution of uncertainties, the optimal controller parameters were calculated using a nonlinear optimization. To avoid an over-conservative design, a suitable uncertainty domain, which expresses the actual type of uncertainty and might not belong to a simple hypercube or interval type, should be considered. On the other hand, the gPC method is suggested only when the number of random parameters is low, e.g. <5 [9]. For higher dimensions, the gPC method with the proper use of a sparse grid should be considered [38], which will be an interesting topic for future studies. Moreover, the idea of shaping the output distribution [39] will be another interesting topic for future research.

Acknowledgments

This study was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012012532).

References

[1] J. Ackermann, Robust Control Systems with Uncertain Physical Parameters, Springer-Verlag, New York, 1993.
[2] S. Skogestad, I. Poslethwaite, Multivariable Feedback Control, John Wiley and Sons, New York, 1996.
[3] S.P. Bhattacharyya, A. Datta, L.H. Keel, Linear Control Theory: Structure, Robustness and Optimization, CRC Press, Boca Raton, 2009.
[4] R. Tempo, G. Caliafore, F. Dabbene, Randomized Algorithm for Analysis and Control of Uncertain Systems: With Applications, 2nd ed., Springer-Verlag, London, 2013.
[5] K.A. Puvkov, N.D. Egupov (Eds.), Classical and Modern Theory of Control System, vol. 2, BMGTU, Moscow, 2003 (in Russian).
[6] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer-Verlag, New York, 2001.
[7] D.P. Kroese, T. Taimre, Z.I. Botev, Handbook of Monte Carlo Methods, John Wiley and Son, Hoboken, NJ, 2011.
[8] H. Niederreiter, P. Hellekalek, G. Larcher, P. Zinterhof, Monte Carlo and Quasi-Monte Carlo Methods, Springer-Verlag, Berlin, 1996.
[9] D. Xiu, Numerical Method for Stochastic Computations: A Spectral Method Approach, Princeton University Press, New Jersey, 2010.
[10] K.A. Puvkov, N.D. Egupov, A.M. Makarenkov, A.I. Trofimov, Theory and Numerical Methods for Studying Stochastic Systems, Fizmatlits, Moscow, 2003 (in Russian).
[11] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Dover Publications, New York, 1991.
[12] D. Xiu, G.R. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2002) 619–644.
[13] P. Li, M. Wendt, G. Wozny, A probabilistically model predictive controller, Automatica 38 (2002) 1171–1176.
[14] G.C. Calafiore, L. Fagiano, Robust model predictive control via scenario optimization, IEEE Trans. Automatic Control 58 (1) (2013) 219–224.
[15] L. Lorefice, B. Pralio, R. Tempo, Randomization-based control design for mini-UAVs, Control Eng. Pract. 17 (2009) 974–983.
[16] G.C. Calafiore, F. Dabbene, R. Tempo, Research on probabilistic methods for control system design, Automatica 47 (2011) 1279–1293.
[17] P.L.T. Duong, M. Lee, Robust PID controller design for process with stochastic parametric uncertainties, J. Process Control 22 (2012) 1559–1566.
[18] Z.K. Nagy, R.D. Braatz, Distributional uncertainty analysis using power series and polynomial chaos expansions Special Issue on Advanced Control of Chemical Processes, J. Process control 17 (2007) 229–240.
[19] Y.T. Tsypkin, B.T. Polyak, Frequency domain for lp robust stability of continuous linear systems, IEEE Trans. Automatic Control 36 (12) (1991) 1464–1469.
[20] B.T. Polyak, P.S. Scherbakov, Random spherical uncertainty in estimation and robustness, IEEE Trans. Automatic Control 45 (11) (2000) 2145–2150.
[21] L. Ljung, System Identification: Theory for User, Prentice Hall, New Jersey, 1987.

[22] M. Barethin, H. Hjalmarsson, Identification and control. Joint input design and H infinity state feedback with parametric uncertainty via LMIs, Automatica 44 (2008) 543–551.
[23] R. Braatz, O. Crisalle, Robustness analysis for system with ellipsoidal uncertainty, Int. J. Robust Nonlinear Control 8 (1998) 1113–1117.
[24] J.A.S. Witteveen, G. Iaccarino, Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces, SIAM J. Sci. Comput. 34 (2012) 814–838.
[25] K.J. Beers, Numerical Methods for Chemical Engineering, Cambridge University Press, New York, 2007.
[26] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, New York, 2004.
[27] R. Zhang, On a transformation method in constructing multivariate uniform designs, Stat. Sin. 6 (1996) 455–469.
[28] I. Podlubny, Fractional Differential Equations, Academic Press, San Diego, 1999.
[29] I. Podlubny, Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers, IEEE Trans. Automatic Control 44 (1) (1999) 208–214.
[30] R. Caponetto, G. Dongola, L. Fortuna, I. Petras, Fractional Order Systems Modeling and Control Applications, World Scientific Publishing Co. Pte. Ltd., Singapore, 2010.
[31] C.A. Monje, Y.Q. Chen, B.M. Vinagre, D.Y. Xue, V. Feliu, Fractional-order Systems and Controls: Fundamentals and Applications, Springer, London, UK, 2010.
[32] J.E. Normey Rico, E.F. Camacho, Control of Dead-time Process, Springer-Verlag, London, 2007.
[33] D.E. Seborg, T.F. Edgar, Process Dynamic and Control, 2nd ed., John Wiley & Son, Hoboken, 2004.
[34] R.L. Magin, Fractional Calculus in Bioengineering, Begell House Publishers Inc., Redding, 2006.
[35] I. Goychuk, P. Hanggi, Fractional diffusion modeling of ion channel gating, Phys. Rev. E 70 (2004) 051915.
[36] Y. Luo, Y.Q. Chen, C.Y. Wang, Y.G. Pi, Tuning fractional order proportional controllers for fractional order system, J. Process Control 20 (2010) 823–831.
[37] R. Malti, M. Aoun, F. Levron, A. Oustaloup, Analytical computational of the H_2 norm of fractional commensurate transfer functions, Automatica 47 (11) (2011) 2425–2432.
[38] P.G. Constantine, M.S. Eldred, E.T. Phipps, Sparse pseudospectral approximation method, Comput. Methods Appl. Mech. Eng. 229–232 (2012) 1–12.
[39] H. Wang, L. Guo, Stochastic Distribution Control System Design: A Convex Optimization Approach, Springer, London, 2010.