

Souransu Nandi

Department of Mechanical Engineering,
University at Buffalo,
Buffalo, NY 14260
e-mail: souransu@buffalo.edu

Victor Migeon

Department of Mechanical Engineering,
University at Buffalo,
Buffalo, NY 14260
e-mail: victor.migeon@orange.fr

Tarunraj Singh

Professor
Department of Mechanical Engineering,
University at Buffalo,
Buffalo, NY 14260
e-mail: tsingh@buffalo.edu

Puneet Singla

Professor
Department of Mechanical Engineering,
University at Buffalo,
Buffalo, NY 14260
e-mail: psingla@buffalo.edu

Polynomial Chaos-Based Controller Design for Uncertain Linear Systems With State and Control Constraints

For linear dynamic systems with uncertain parameters, design of controllers which drive a system from an initial condition to a desired final state, limited by state constraints during the transition is a nontrivial problem. This paper presents a methodology to design a state constrained controller, which is robust to time invariant uncertain variables. Polynomial chaos (PC) expansion, a spectral expansion, is used to parameterize the uncertain variables permitting the evolution of the uncertain states to be written as a polynomial function of the uncertain variables. The coefficients of the truncated PC expansion are determined using the Galerkin projection resulting in a set of deterministic equations. A transformation of PC polynomial space to the Bernstein polynomial space permits determination of bounds on the evolving states of interest. Linear programming (LP) is then used on the deterministic set of equations with constraints on the bounds of the states to determine the controller. Numerical examples are used to illustrate the benefit of the proposed technique for the design of a rest-to-rest controller subject to deformation constraints and which are robust to uncertainties in the stiffness coefficient for the benchmark spring-mass-damper system. [DOI: 10.1115/1.4038800]

1 Introduction

It is well known that the performance of open-loop control strategies for motion control of flexible structures is sensitive to imperfect models. This has prompted numerous researchers to investigate the problem of developing controllers that are robust to parametric uncertainties in the models [1–6]. One approach has been to design controllers to take care of worst case scenarios among all realizations, which corresponds to a minimax design which requires knowledge of the support of the uncertainties. A second strategy of dealing with parametric uncertainties in the model has been to reduce the sensitivity of the cost function in the proximity of the nominal model by forcing the local sensitivity to zero.

The aforementioned problem formulations have often considered minimizing the residual energy, i.e., the undesired energy in the system at the end of the maneuver. This is important in rest-to-rest class of problems. In addition to the residual energy constraint, deflection constraints have been considered by Singhose et al. [7] where an analytical expression for the evolution of the deflection for a spring-mass system is derived and its gradient is forced to zero to identify time instants, which correspond to maximum or minimum deflections. Deflection sampling is also used to constrain the permitted deflection in the design of the controller. Vossler and Singh [8] addressed the problem of design of the optimal control profiles for systems with uncertain parameters, for rest-to-rest maneuvers with state constraints. They formulated a linear programming (LP) problem with a cost function of minimizing the maximum excursion of the states over the duration of the maneuver called the *minimax control*. This, in conjunction with constraints on the terminal energy, resulted in a robust state and energy constrained controller. A uniform sampling of the uncertain space was used to solve the minimax problem. Monte Carlo (MC) realization can also be used to sample the uncertain space. However, it is well known that MC methods suffer from slow convergence with convergence rate being inversely

proportional to the square root of number of sample points. This means that one needs to increase the number of samples by 100 to improve the accuracy by one decimal place. Furthermore, there is no guarantee that model parameters, which span the complement of the finite samples, will not violate the state and energy constraints. This paper considers applications where constraints have to be imposed on states over the duration of the maneuver, in the presence of model parameter uncertainties. The proposed approach does not depend on sampling to identify the realization(s), which correspond to the active constraints.

Building on the rich literature of the use of polynomial chaos (PC) in control [9–13], Nandi et al. [14] presents an approach, which rewrites the polynomials associated with the specified distributions of the uncertain variables using Bernstein polynomials. The mapping permits determination of time evolutions of the bounds on the uncertain states and is illustrated on a numerical example with one uncertain parameter. This paper generalizes the development in Ref. [14] and presents a systematic technique to generate tight bounds on the evolving states which are required for the controller design. In addition, since the illustrative examples in the paper include uniformly distributed uncertain parameters, comparisons to interval analysis (IA) methods are also presented. It is observed that bounds determined from IA are severely conservative making the Bernstein formulation a more appropriate choice. Then, an optimization problem is posed to ensure that the state constraints are not violated for all possible realizations of the uncertain variables. The proposed approach is attractive since it does not require Monte Carlo simulations to estimate the bounds on the states. Finally, an LP problem can be posed to determine a controller, which is robust to the uncertain variables. Although the individual ideas of a PC expansion of stochastic states, the Bernstein formulation to bound polynomials and control input design methodology via linear programming is not new, the novelty of the work lies in the fluid integration of the three frameworks, which has not been studied before.

The structure of the paper is as follows: In Sec. 2, an LP-based approach for the design of optimal control profiles, which are robust to model uncertainties, is presented. This motivates the need for the development of a PC-based approach for state constrained control for uncertain systems. A review of PC is provided

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received May 3, 2017; final manuscript received December 13, 2017; published online January 24, 2018. Assoc. Editor: Douglas Bristow.

in Sec. 3 to characterize uncertainties and propagate them through the dynamical model. This is followed in Sec. 4 by the PC mapping using Bernstein polynomial basis function, which illustrates how the bounds on the states are determined. Section 5 presents the bounds determined from IA and provides motivation for using the Bernstein method over IA. Section 6 elaborates on the improvement on these bounds by splitting the Bernstein coefficients. Section 7 illustrates the robust controller design using LP and the numerical results corresponding to the benchmark floating oscillator problem.

2 Controller Design

This section describes the linear programming-based approach for robust optimal control by following the development in Ref. [15]. The problem is formulated for rest-to-rest maneuvers of vibratory system in the presence of model uncertainties using the minimax approach [8], where the inclusion of state constraints into the design results in a much more challenging design problem. The resulting problem statement for state-limited control is

$$\min_{u(t)} F = [\max_{\alpha} E(\alpha)] \quad (1a)$$

$$\text{subject to } \dot{x}(t) = A_c x(t) + B_c u(t) \quad (1b)$$

$$x(t_0) = x_0; \quad x(t_f) = x_f \quad (1c)$$

$$u_{lb} \leq u(t) \leq u_{ub} \quad \forall t \quad (1d)$$

$$\text{state constraints } \forall t \quad (1e)$$

where $x(t) = [p^T, v^T]^T \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^k$, u_{lb} , and u_{ub} are lower and upper bounds on the control input, respectively. $E(\alpha)$ is the residual energy given by

$$E(\alpha) = \frac{1}{2} v^{(\alpha)T} M v^{(\alpha)} + \frac{1}{2} [p^{(\alpha)} - p_{ref}]^T K(\alpha) [p^{(\alpha)} - p_{ref}] \quad (2)$$

where M is the mass matrix, $K(\alpha)$ is the uncertain stiffness matrix, $p^{(\alpha)}$ and $v^{(\alpha)}$ are the positional and velocity states corresponding to a realization of the uncertain α , respectively. α represents a finite parametric set of the uncertain variable that must lie within the domain of uncertainty specified by

$$\alpha^{lb} \leq \alpha \leq \alpha^{ub} \quad (3)$$

where α^{lb} and α^{ub} represent the lower and upper bounds, respectively. The nonlinear optimization problem can be transformed into a convex problem in the discrete domain by posing it as a linear programming problem [15].

Consider the discrete time realization of Eq. (1b)

$$x(k+1) = Ax(k) + Bu(k) \quad (4)$$

The recursive relation of Eq. (4) can be expressed as a linear expression of the initial conditions and the control input u as

$$x(k+1) = A^k x(1) + \sum_{i=1}^k A^{k-i} B u(i) \quad (5)$$

where $x(1)$ is the initial condition of the states.

The objective of the problem is to determine a control input u , which drives the dynamical system from an initial state $x(1)$ to a final state $x(N_t + 1)$, where $k = N_t$ is the final iterative step of Eq. (4) in a given finite interval of time. Thus, the known quantities of the problem are initial time (T_0), final time (T_f), initial conditions ($x(1)$), and final conditions ($x(N_t + 1)$).

The LP problem is posed as a feasibility problem in this case with linear constraints. From Eq. (5), the final condition constraint is given by

$$\begin{bmatrix} A^{N_t-1}B & A^{N_t-2}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} u(1) \\ u(2) \\ \vdots \\ u(N_t) \end{bmatrix} = x(N_t + 1) - A^{N_t}x(1) \quad (6)$$

State constraints can be implemented by

$$CA^k x(1) + \sum_{i=1}^k CA^{k-i} B u(i) \leq \Phi \quad \forall k \in [1: N_t] \quad (7)$$

where C is the output matrix and Φ represents a prespecified bound. The control constraints are implemented as

$$u_{lb} \leq u(k) \leq u_{ub} \quad \forall k \in [1: N_t] \quad (8)$$

where u_{lb} and u_{ub} are lower and upper bounds on the control inputs. The solution to the problem stated above yields the desired control input over all the time instants ($k = [1: N_t]$).

Singh [16] proposed a linear programming-based approach for the design of optimal controllers, which are robust to model parameter uncertainties. This was accomplished by uniformly sampling the compact support of the uncertainty. A minimax problem is now formulated to minimize the worst residual energy at the end of the maneuver over the sampled models. It was shown that the resulting solution closely approximated the solution generated by the nonlinear programming problem.

When state constraints are included in the controller design problem, a large number of additional constraints are included in the linear programming problem. These constraints correspond to the satisfaction of the state constraint at every time instant. Since the accuracy of the optimal control improves as the sampling time decreases, one needs to tradeoff the computational cost of the additional state constraints to the improved representation of the optimal control. When one endeavors to solve the state-constrained optimal control problem in the presence of model uncertainties, the problem quickly becomes intractable as the number of uncertainties grows. To address this issue, a polynomial chaos-based approach is proposed in this paper.

3 Polynomial Chaos Expansion

Polynomial chaos was first introduced by Norbert Wiener in 1938 in his article [17]. In this work, he first approximated a stochastic state following a Gaussian process by an infinite series expansion of orthogonal Hermite polynomials. Later, in 1947, Cameron and Martin [18] proved that such an expansion always converges for stochastic processes with a finite variance. Years later, this property was used by Ghanem and Spanos in their book [19] to solve stochastic differential equations. Instead of an infinite expansion, they truncate the series to a finite number of members. Then they use Galerkin projection to formulate a set of deterministic equations, and finally solve them to obtain the coefficients of their original series expansion. Xiu and Karniadakis [20] generalize the concept of PC to express the generic stochastic process as a series expansion of appropriate orthogonal polynomials (given by the Wiener-Askey scheme). This development was termed as the generalized polynomial chaos (gPC) theory. This particular concept is used in the present work to approximate a stochastic process with uniformly distributed variables.

A detailed formulation of the PC expansion is presented in this section and illustrated on a benchmark problem.

3.1 Methodology. Let a stochastic dynamical system be expressed in the form

$$\dot{\mathbf{x}}(t, \xi) = \mathbf{f}(\mathbf{x}(t), \xi, \mathbf{u}(t)) \text{ and } \mathbf{x}(t_0, \xi) = \mathbf{x}_0 \quad (9)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\xi \in \mathbb{R}^m$ is the vector of random variables with known probability distributions, $\mathbf{x}_0 \in \mathbb{R}^n$ is the initial state (which may also be uncertain), and $\mathbf{u}(t)$ is the control input.

From the theory of generalized gPC, the states can be expressed as

$$\mathbf{x}(t, \xi) = \sum_{i=0}^{\infty} \mathbf{x}_i(t) \Psi_i(\xi) \quad (10)$$

where $\Psi_i(\xi)$ is a complete set of multivariate orthogonal polynomials depending on the type of distribution of ξ and $\mathbf{x}_i \in \mathbb{R}^n$ is the time varying coefficient vector of the polynomials $\Psi_i(\xi)$.

The selection of the set of orthogonal polynomials for popular distributions is given by the Wiener–Askey scheme [20]. The expansion is typically truncated to a finite number of terms as an approximation of the expression. The number of terms to be included in the approximation is determined by the degree of accuracy desired. Larger the number of terms (N), longer the time span over which the states are accurate [20]. Hence, Eq. (10) is rewritten as

$$\mathbf{x}(t, \xi) = \sum_{i=0}^N \mathbf{x}_i(t) \Psi_i(\xi) \quad (11)$$

which is a vector equation

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} x_{10} \\ x_{20} \\ \vdots \\ x_{n0} \end{bmatrix}}_{\mathbf{x}_0} \Psi_0 + \underbrace{\begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix}}_{\mathbf{x}_1} \Psi_1 + \cdots + \underbrace{\begin{bmatrix} x_{1N} \\ x_{2N} \\ \vdots \\ x_{nN} \end{bmatrix}}_{\mathbf{x}_N} \Psi_N \quad (12)$$

where x_i is the i th state and x_{ij} is the coefficient of the j th polynomial belonging to the i th state.

The objective is to determine all the time varying coefficients, i.e., x_{ij} . The basis functions of the states, i.e., Ψ_i , are a set of orthogonal polynomials with respect to the probability density function (pdf) of the random variable. If a system has multivariate independent random variables, the basis functions are the multivariate polynomials derived from the tensor product of the univariate basis functions of each random variable. To continue with the PC formulation, Eq. (11) is substituted in Eq. (9) to get

$$\sum_{i=0}^N \dot{\mathbf{x}}_i(t) \Psi_i(\xi) = \mathbf{f}\left(\sum_{i=0}^N \mathbf{x}_i(t) \Psi_i(\xi), \xi, \mathbf{u}(t)\right) \quad (13)$$

The essence of PC expansion is to form a set of deterministic differential equations from the stochastic equation (13), whose solution allows us to approximate the states over time. These equations are formed by performing the Galerkin Projection on it over each of the orthogonal basis functions (i.e., Ψ_k , where $k=0, 1, \dots, N$). The solution to these equations yields our desired coefficients x_{ij} .

The k th deterministic differential equation on taking the Galerkin Projection is given by

$$\left\langle \sum_{i=0}^N \dot{\mathbf{x}}_i(t) \Psi_i(\xi), \Psi_k(\xi) \right\rangle = \left\langle \mathbf{f}\left(\sum_{i=0}^N \mathbf{x}_i(t) \Psi_i(\xi), \xi, \mathbf{u}(t)\right), \Psi_k(\xi) \right\rangle \quad (14)$$

The property of orthogonality of the polynomials, permits writing the inner product as

$$\langle \Psi_i(\xi), \Psi_j(\xi) \rangle = \int_{\Omega} \Psi_i(\xi) \Psi_j(\xi) pdf(\xi) d\xi = c_i \delta_{ij} \quad (15)$$

for $k=0, 1, \dots, N$ where δ_{ij} is the Kronecker delta function, c_i is a coefficient whose value depends on the orthogonal polynomials, and Ω is the support of the distribution of the random variables. Now, Eq. (14) can be simplified to yield

$$\dot{\mathbf{x}}_k(t) = \frac{1}{c_k} \int_{\Omega} \mathbf{f}\left(\sum_{i=0}^N \mathbf{x}_i \Psi_i(\xi), \xi, \mathbf{u}(t)\right) \Psi_k(\xi) pdf(\xi) d\xi \quad (16)$$

for $k=0, 1, \dots, N$.

Equation (16) is a set of $(N+1)$ vector differential equations with the vectors having n elements each. Hence, a total of $(N+1)n$ scalar equations need to be solved simultaneously. Since the equations are deterministic, they can be easily numerically evaluated.

3.2 Illustrative Example. To clearly demonstrate the theory presented above, an implementation of it on a simple two mass spring damper system (Fig. 1) is undertaken. PC expansion is used to determine the stochastic states, and the results are compared with MC simulations.

The first and the second bodies have masses m_1 and m_2 with displacements y_1 and y_2 , respectively. The coefficient of the spring between them is k and the damping constant is c . The deterministic control input $u(t)$ is applied to the first mass.

Assuming that the spring constant is uncertain with a uniform distribution

$$k = \mathcal{U}(0.7, 1.3) \quad (17)$$

the Wiener–Askey scheme, for a uniform distribution, requires the orthogonal polynomials of expansion to be Legendre polynomials. The orthogonality of the polynomials is present in the domain $[-1, 1]$. Hence, the spring constant (k) is defined in terms of another random variable (ξ) with a similar domain, i.e., $\xi = \mathcal{U}(-1, 1)$ and thus, we have

$$k = 1 + 0.3\xi \quad (18)$$

Truncating the PC expansion to the fifth-order, i.e., $N=5$, the states are expanded as

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_{10} \\ y_{20} \\ y_{30} \\ y_{40} \end{bmatrix} \Psi_0 + \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{41} \end{bmatrix} \Psi_1 + \cdots + \begin{bmatrix} y_{15} \\ y_{25} \\ y_{35} \\ y_{45} \end{bmatrix} \Psi_5 \quad (19)$$

and the dynamic system in terms of the uncertain variable is

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{(1+0.3\xi)}{m_1} & \frac{(1+0.3\xi)}{m_1} & -\frac{c}{m_1} & \frac{c}{m_1} \\ \frac{(1+0.3\xi)}{m_2} & -\frac{(1+0.3\xi)}{m_2} & \frac{c}{m_2} & -\frac{c}{m_2} \end{bmatrix}}_A \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix}}_B u(t) \quad (20)$$

Differentiating Eq. (19) with respect to time and equating it to Eq. (20), we get four equations

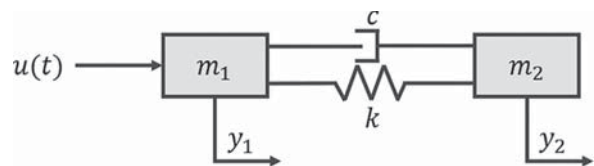


Fig. 1 Two mass spring damper system

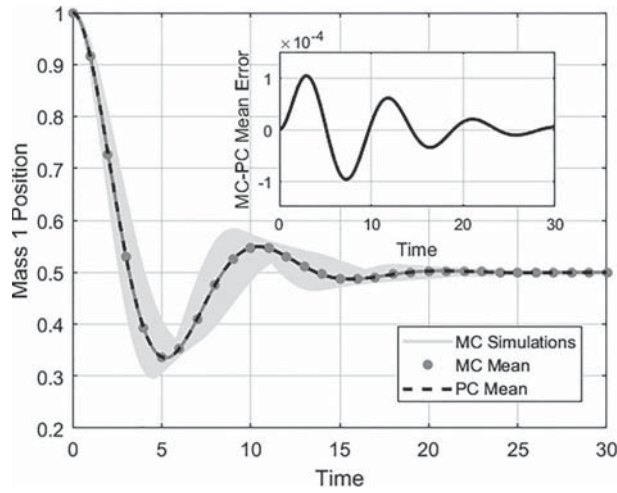


Fig. 2 MC simulations (10,000), MC mean and PC mean for mass 1 position—one uncertain variable

$$\begin{aligned} \dot{y}_{10}\Psi_0 + \dot{y}_{11}\Psi_1 + \cdots + \dot{y}_{15}\Psi_5 &= A_{11}(y_{10}\Psi_0 + \cdots + y_{15}\Psi_5) \\ &+ A_{12}(y_{20}\Psi_0 + \cdots + y_{25}\Psi_5) + \cdots + A_{14}(y_{40}\Psi_0 + \cdots + y_{45}\Psi_5) \\ &+ B_{11}u(t) \end{aligned} \quad (21)$$

⋮

$$\begin{aligned} \dot{y}_{40}\Psi_0 + \dot{y}_{41}\Psi_1 + \cdots + \dot{y}_{45}\Psi_5 &= A_{41}(y_{10}\Psi_0 + \cdots + y_{45}\Psi_5) \\ &+ A_{42}(y_{20}\Psi_0 + \cdots + y_{25}\Psi_5) + \cdots + A_{44}(y_{40}\Psi_0 + \cdots + y_{45}\Psi_5) \\ &+ B_{41}u(t) \end{aligned} \quad (22)$$

where A_{ij} and B_{ij} are the i th row- j th column elements of A and B , respectively. Taking the Galerkin projection on Eq. (21), we get $(N+1) = 6$ deterministic equations

$$\dot{y}_{10}\langle\Psi_0, \Psi_0\rangle = f_{10}(y_{10}, \dots, y_{4N}, u) \quad (23)$$

⋮

$$\dot{y}_{15}\langle\Psi_5, \Psi_5\rangle = f_{15}(y_{10}, \dots, y_{4N}, u) \quad (25)$$

Similarly, Galerkin projections on the other equations from Eqs. (21) and (22) yield a total of $((N+1)n=24)$ differential equations. These equations are solved to determine all the coefficients of the PC expansion representation of the states.

Simulating the unforced system with nonuncertain system parameters of $c=1$, $m_1=5$, $m_2=5$, and initial conditions $y(0) = (1, 0, 0, 0)^T$, Fig. 2 illustrates the time evolution of the mean of the position of the first mass evaluated using PC expansion (truncated to the fifth-order, i.e., $N=5$). It is plotted along with Monte Carlo simulations and the mean is derived from 10,000 such simulations. The overlapping curves for the mean of the Monte Carlo and the mean determined from PC prove the convergence of PC expansion over the given interval of time. The inset graph is an illustration of the difference between the mean estimated from Monte Carlo simulations and the mean estimated from the PC expansion, clearly illustrating the small difference in the two estimates. The higher-order moments can also be similarly compared.

To illustrate the development of the PC model for two uncertain variables, consider the mass spring damper system with the following uncertainties:

$$k = \mathcal{U}(0.7, 1.3); \quad c = \mathcal{U}(0.8, 1.2) \quad (26)$$

Following the same argument as before, new random variables are defined as:

$$\xi_1 = \mathcal{U}(-1, 1); \quad \xi_2 = \mathcal{U}(-1, 1) \quad (27)$$

leading to

$$k = 1 + 0.3\xi_1; \quad c = 1 + 0.2\xi_2 \quad (28)$$

Since there are two independent variables, the basis functions (orthogonal polynomial functions) are now functions derived from the tensor product of univariate polynomial functions. In this case, since both the uncertain variables have uniform distribution, the product is between Legendre polynomials. Numerical simulation of the gPC model model for two uncertain variables confirmed the agreement with the mean estimated from 10,000 Monte Carlo runs.

4 Determination of Bernstein Bounds

In numerous fields of engineering, it is often desired to determine the bounds on the range of a particular state or function. If the particular function of interest is, or can be well approximated by a multivariate polynomial, Bernstein polynomials can be exploited to determine these bounds [21]. In fact, algorithms have also been proposed to determine the exact range of multivariate polynomials using Bernstein expansions [12,22]. The work presented here, however, makes use of the bounding properties of Bernstein bases to estimate tight bounds on the range of stochastic states and use these bounds as constraints to design a robust controller.

In Sec. 3, it was shown that gPC allows a good approximation of states of a stochastic process, as a polynomial function of the random variables. The example in particular had Legendre polynomials as the basis functions. This section highlights a procedure to find those bounds and improve them. As bounds only exist for compact support, the procedure only works when variables have a finite distribution.

Bounds' determination is done by a basis transformation from the existing bases (of PC expansion) to the Bernstein bases. The transformation allows the exploitation of Bernstein properties to establish bounds. Garloff [23] shows that these bounds can be improved if the domain of the Bernstein polynomials is subdivided. This subdivision to improve bounds is done using De Casteljau's algorithm. The algorithm can be found in Ref. [24].

4.1 Methodology. A polynomial function can be expressed in terms of other polynomial bases. The most common form used is the power bases, i.e.,

$$1, \quad \xi, \quad \xi^2, \quad \xi^3, \dots \quad (29)$$

Other popular forms include the Bernstein basis or the set of orthogonal polynomials (like Legendre, Jacobi, Hermite, etc). Polynomials whose orthogonality domain is finite (Legendre, shifted Legendre, Jacobi,...) are used as bases in PC expansion of stochastic states with compact support. Hence, the first objective is a basis transformation from the orthogonal bases to the Bernstein bases.

A stochastic state expressed as a PC expansion

$$x_j(t, \xi) = \sum_{i=0}^N x_{ji}(t) \Psi_i(\xi) \quad (30)$$

where x_j is the j th state of the model, Ψ_i are the multivariate orthogonal bases (Legendre, Jacobi,...), and $x_{ji}(t)$ are the corresponding coefficients of the bases now need to be expressed as

$$x_j(t, \xi) = \sum_{i=0}^N b_{ji}(t) \mathcal{B}_i(\xi) \quad (31)$$

where $\mathcal{B}_i$ are the multivariate Bernstein polynomials with domain $\xi \in [0, 1]^m$ and $b_{ji}(t)$ are their coefficients. This linear transformation (from Eqs. (30) to (31)) of the coefficients involves three stages.

The first stage requires Ψ_i (which has domain of orthogonality $[a, b]$) to be transformed to orthogonal bases Ψ'_i with orthogonality domain $[0, 1]$ as the property of Bernstein polynomials to be utilized is satisfied only within that domain. Hence, the first transformation is given by

$$x_j(t, \xi) = \sum_{i=0}^N x_{ji}(t) \Psi_i(\xi) \rightarrow x_j(t, \xi) = \sum_{i=0}^N x'_{ji}(t) \Psi'_i(\xi) \quad (32)$$

The second stage requires the transformation from Ψ'_i to the power bases P_i

$$x_j(t, \xi) = \sum_{i=0}^N x'_{ji}(t) \Psi'_i(\xi) \rightarrow x_j(t, \xi) = \sum_{i=0}^N p_{ji}(t) P_i(\xi) \quad (33)$$

The final transformation gives the desired polynomial expressed in terms of multivariate Bernstein bases

$$x_j(t, \xi) = \sum_{i=0}^N p_{ji}(t) P_i(\xi) \rightarrow x_j(t, \xi) = \sum_{i=0}^N b_{ji}(t) \mathcal{B}_i(\xi) \quad (34)$$

Equations (32)–(34) refer to linear transformations and vary according to the number of random variables, the original bases functions, and the order of truncation. For example, Legendre–Bernstein transformations and Jacobi–Bernstein transformation methods are provided by Farouki [25] and Rababah [26], respectively.

Multivariate Bernstein polynomial bases are derived from the tensor product of univariate Bernstein polynomials. Without loss of generality, the subscript j is dropped (to indicate any state member) for convenience from Eq. (31), and the equation is rewritten in terms of Bernstein polynomials as

$$x(t, \xi) = \sum_{\xi_1, \xi_2, \dots, \xi_m=0}^{d, d, \dots, d} b_{\xi_1, \xi_2, \dots, \xi_m}(t) B_{\xi_1, \xi_2, \dots, \xi_m}^d(\xi) \quad (35)$$

where $b_{\xi_1, \xi_2, \dots, \xi_m}$ are the Bernstein coefficients, $B_{\xi_1, \xi_2, \dots, \xi_m}^d(\xi) = \prod_{i=1}^m B_{\xi_i}^d(\xi_i)$ are the m -dimensional Bernstein polynomials and d is the degree of the univariate polynomials. Now, the range enclosing property of Bernstein polynomials over the box $\xi \in [0, 1]^m$

$$x(t, \xi) \subseteq \left[\min_{\xi_1, \xi_2, \dots, \xi_m=0}^{d, d, \dots, d} b_{\xi_1, \xi_2, \dots, \xi_m}(t), \max_{\xi_1, \xi_2, \dots, \xi_m=0}^{d, d, \dots, d} b_{\xi_1, \xi_2, \dots, \xi_m}(t) \right] \quad (36)$$

is used to immediately obtain bounds on the state expressions.

Equation (36) can also be expressed as a convex hull property by defining control points

$$\left\{ \begin{pmatrix} \xi \\ x(t, \xi) \end{pmatrix} : \xi \in [0, 1]^m \right\} \subseteq \text{conv} \left\{ \begin{pmatrix} (b-a) * \left(\frac{\xi_1}{d}, \frac{\xi_2}{d}, \dots, \frac{\xi_m}{d} \right) \\ b_{\xi_1, \xi_2, \dots, \xi_m}(t) \end{pmatrix} : \xi_1, \xi_2, \dots, \xi_m = 0, 0, \dots, 0 \text{ to } d, d, \dots, d \right\} \quad (37)$$

where $\text{conv}\{M\}$ denotes the convex hull of set M .

4.2 Illustrative Example. Solution to Eqs. (23)–(25) gives the PC expansion of the first state. Analysis of just this one state is shown henceforth, i.e., the position of mass 1. Analysis for the other states can also be done in a similar manner

$$y_1 = y_{10} \Psi_0 + y_{11} \Psi_1 + \dots + y_{15} \Psi_5 \quad (38)$$

The Legendre coefficients $[y_{10}, \dots, y_{15}]^T$ are transformed to Bernstein coefficients $[b_{10}, \dots, b_{15}]^T$ using the methodology described in Ref. [25] where a transformation matrix is developed to transform multivariate Legendre polynomial coefficients to multivariate Bernstein ones based on Eqs. (32)–(34). Therefore, y_1 is expressed as

$$y_1(t) = b_{10}(t) B_0^5(\xi) + \dots + b_{15}(t) B_5^5(\xi) \quad (39)$$

Using the property mentioned in Eq. (36), at every instant in time, the maximum and minimum values of the state are obtained directly by observing the coefficients $[b_{10}, \dots, b_{15}]^T$. The maximum values of the coefficients provide the upper bound while the minimum values provide the lower bound. Figure 3(a) shows that the state values determined from all the MC simulations of the model lie within the envelope provided by the bounds.

Slices taken from Fig. 3(a) at times $t = 11, 17$, and 29 are shown in Figs. 4(a)–4(c). These time instants were selected to show a variety of convex hulls. The stars denote the control points derived from Eq. (37). The convex hull determined from the set of control points form a superset for all points lying on the black curve (curve of mass 1 position with varying k). Hence, finally, a deterministic bound is now available for the states of a stochastic process.

Similar to the one-dimensional (1D) case (one uncertain variable), a transformation can be used to convert the multivariate Legendre coefficients to Multivariate Bernstein coefficients [25]. The definitive bounds once again directly drop out of the coefficients and have been shown in Fig. 3(b).

Slices are once again taken, at times $t = 17, 19$, and 27 . Since there are two uncertain parameters now, the variation of the state is shown as a surface plot. Using relation (37), new control points are generated. A convex hull is determined from the set of these control points and is shown to envelope the entire surface of the states (Figs. 5(a)–5(c)), thus, proving the validity of the bounds.

5 Determination of Interval Analysis Bounds

Interval analysis, first introduced by Moore [27], represents any real number (x) by an interval bounded by a lower limit and an upper limit (i.e., $x = [\underline{x}, \bar{x}]$). All mathematical operations between such quantities are done in terms of their intervals. For example, if $x = [\underline{x}, \bar{x}]$ and $y = [\underline{y}, \bar{y}]$, then $x + y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$. Similar definitions for difference, multiplication, etc., are present, with the guarantee that the output variable will lie within the resulting interval.

For linear time invariant systems (class of systems being studied), parametric uncertainties can be mapped to an interval system matrix $[A]$ where every element of $[A]$ is an interval variable. The solution to an unforced linear time invariant system with nonuncertain initial conditions and an interval system matrix $[A]$ is given by: $x(t) = e^{[A]t} x(0)$. This expression requires the evaluation of another interval matrix $e^{[A]t}$ at every instant t , which is typically approximated by a truncated Taylor series expansion and a bounding truncation error [28].

Interval analysis suffers from the major drawback of overestimation of true intervals. In fact, calculating the true bounds of the interval matrix exponential ($e^{[A]t}$) is a NP-hard problem [29]. Althoff et al. [28] calculates the exponential interval by determining exact bounds on the first three terms of the Taylor series expansion and overestimating the rest of the series (method 1).

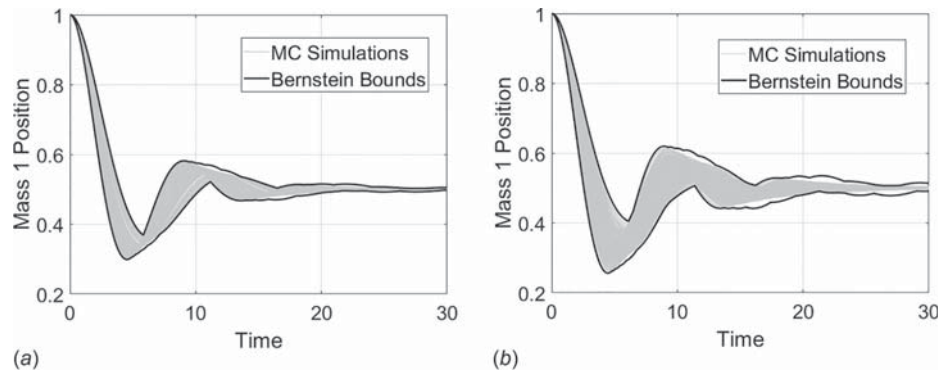


Fig. 3 Bernstein bounds: (a) definite bounds from Bernstein coefficients (1D) and (b) definite bounds from Bernstein coefficients two-dimensional (2D)

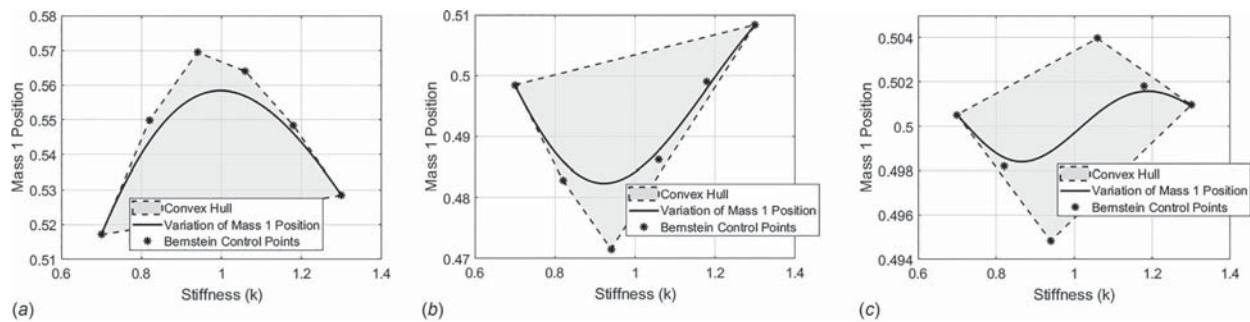


Fig. 4 Convex hulls: (a) time = 11, (b) time = 17, and (c) time = 29

Truncation error is also considered. An improvement to this method is seen by re writing the Taylor expansion in the Horner format of polynomials accompanied by a scaling-squaring process (method 2) [29].

Both these methods allow determination of bounds in a continuous domain. An extension can be easily made to the discrete one. The motivation to do so lies in the fact that determining the matrix exponential at every instant in time can be computationally significantly expensive. In discrete domain, however, the matrix exponential needs to be evaluated just once. For a given sampling time T , the discrete state transition matrix ($e^{[A]T}$) is calculated using similar methods as the continuous domain. Once the interval state transition matrix is known, the interval states are propagated using the rules of interval arithmetic matrix–vector product.

The bounds can be further improved if the entire interval is subdivided and the results from the analysis of each separate smaller interval are combined [30].

Figure 6 shows a comparison of the nature of bounds determined from each of these methods as well as the ones determined

from Bernstein coefficients. Simulation parameters are the same as Sec. 3.2 with the stiffness interval $k = [0.7, 1.3]$. For the simulation, the stiffness interval was divided into four equal subintervals: $[0.7, 0.85]$; $[0.85, 1]$; $[1, 1.15]$ and $[1.15, 1.3]$ and bounds were combined from each interval. Continuous method 1 and both discrete methods (sampling time: $T = 0.01$) quickly become very conservative and thus have not been shown beyond times $t = 4$ and $t = 6$, respectively. Continuous method 2 performs the best among all the IA techniques. However, bounds from Bernstein coefficients are far superior. Hence, it makes sense to adopt the Bernstein bounds over the IA techniques for subsequent work.

6 Splitting Bernstein Coefficients

Definite bounds were established from the extreme values of the Bernstein coefficients in Sec. 4. However, Figs. 4(a)–5(c) make a visual point toward stating that the bounds yet have room for improvement. Existing bounds can be made tighter by splitting the Bernstein coefficients along each dimension.

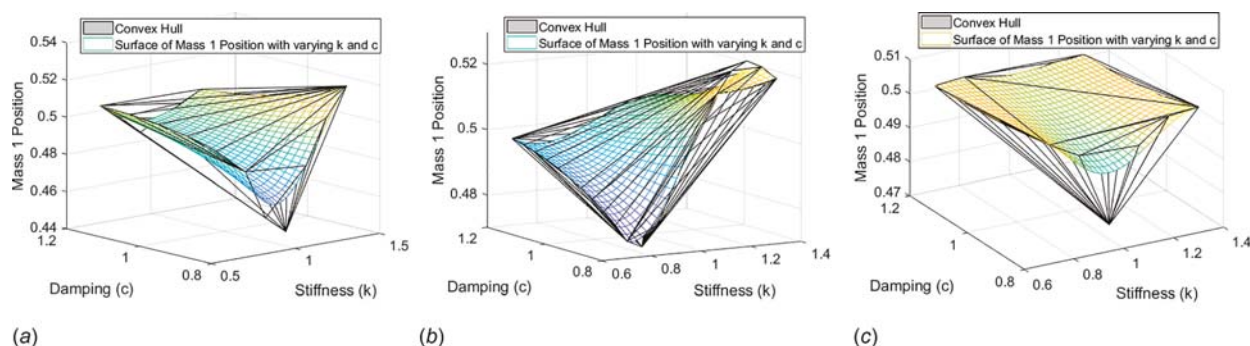


Fig. 5 Three-dimensional convex hulls: (a) time = 17, (b) time = 19, and (c) time = 27

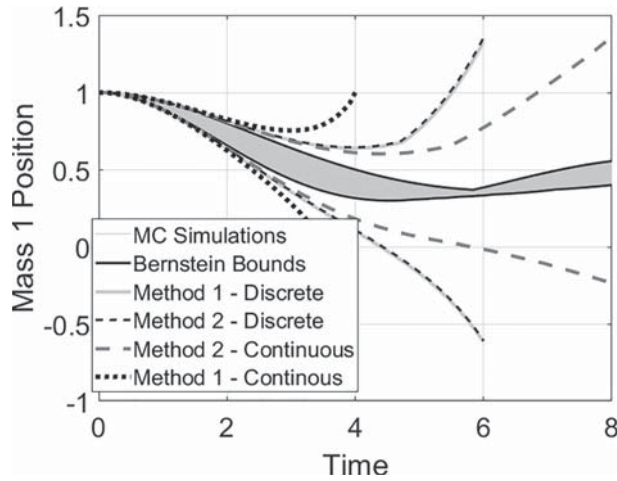


Fig. 6 Comparison of bounds

Splitting of Bezier curves (parametric curves which are determined by Bernstein polynomials and lie within the convex hull of its control points) was first presented by French engineer de Casteljau [31]. His algorithm (famously known as de Casteljau's algorithm in computer graphics) is used to date to split the convex hull into segments. In our case, that reduces the conservative nature of the bounds.

Figure 4(c) (pertaining to example 1) shows a Bezier curve with its control points marked with asterisk. De Casteljau's algorithm is used to split this Bezier curve into two separate curves each with its own set of control points. The split can be made at any arbitrary point ($\xi = a$; $0 \leq a \leq 1$).

The procedure for problems with one uncertain variable (1D) is explained prior to making the procedure more generic.

Consider the 1D control points (k, b_{1i}) from Fig. 4(c), i.e., the Bezier curve at $t = 29$ (Table 1):

The lower and upper bounds for this curve is given by

$$\min(b_{1i}) = 0.4948 \quad \max(b_{1i}) = 0.5040 \quad (40)$$

Hence, the lower bound and upper bound for mass 1 position at time = 29 is 0.4948 and 0.5040, respectively.

On applying de Casteljau's algorithm, and splitting the curve at

$$\xi = a = 0.5; \quad \text{i.e.,} \quad k = 0.7 + (1.3 - 0.7)a = 1 \quad (41)$$

the new control points (Fig. 7(a)) obtained are listed in Table 2.

The lower and upper bounds for this curve is now given by

$$\min(b_{1i}) = 0.4979 \quad \text{and} \quad \max(b_{1i}) = 0.5021 \quad (42)$$

The lower bound for mass 1 position increases and the upper bound decreases, thus improving the definitive bound. In this particular case, the range on the bounds see a decrease of 54.13% when the coefficients are split, indicating a substantial improvement.

Splitting of Bernstein surfaces or higher dimensional coefficients is slightly more involved than the 1D case. An explanation of the 2D case makes it simpler to understand the procedure for higher dimensional cases. The following discussion deals with the splitting of the Bernstein surface represented in Fig. 5(c).

Table 1 1D unsplit control points

k	b_{1i}	k	b_{1i}	k	b_{1i}
0.7000	0.5005	0.8200	0.4982	0.9400	0.4948
1.0600	0.5040	1.1800	0.5018	1.3000	0.5010

Since, now we have a Bernstein surface (2D), the control points must be laid out in the form of a meshgrid (Table 3). Unsplit Bernstein coefficients at $t = 27$ (Table 3) shows all the Bernstein coefficients with horizontally varying ξ_1 (i.e., k) and vertically varying ξ_2 (i.e., c).

The bounds of the state are calculated by observing the maximum and the minimum value of the coefficients (from Table 3) where

$$\min(b_{\xi_1, \xi_2}) = 0.4779 \quad \text{and} \quad \max(b_{\xi_1, \xi_2}) = 0.5068 \quad (43)$$

Hence, the lower and upper bounds of the state at time = 27 are 0.4779 and 0.5068, respectively, for unsplit control points.

Using De Casteljau's algorithm, the coefficients are split along $\xi_1 = 0.5$ (i.e., $k = 1$) first to get the partially split control points. The coefficients so obtained are now further split; however, this time along $\xi_2 = 0.5$ (i.e., $c = 1$) to obtain completely split coefficients (Table 4).

Bounds are determined in the same way from Table 4. The lower bound found is 0.4922 and the upper bound is 0.5068. Again, we see that there is an improvement in the bounds once the coefficients are split (Split bounds improve the range by 49.52%). The convex hull formed from the split control points is shown in Fig. 7(b) as opposed to unsplit control points in Fig. 5(c). For higher dimensional cases, the procedure for the 2D case is followed. The splitting is carried out separately along each dimension independently. The bounds as well as the convex hull are determined after the split control points (i.e., split Bernstein coefficients) have been calculated.

It should be noted that it is not necessary that the splitting be done along all the dimensions. Splitting, however, should be carried out along dimensions, which have the most influence over the states to substantially improve bounds.

Moreover, throughout the procedure, the splitting has been done once and for all cases at $\xi = 0.5$. Splitting the coefficients more than once is possible and does improve bounds. However, splitting the coefficients more than once greatly increases computational requirements considering that the computations have to be performed at all times. Splitting can also be done at values other than $\xi = 0.5$ and would yield different results. It would be ideal if the splitting could be done at an optimal point; however, determining the optimal point at every stage is once again expensive and a computational compromise is made.

7 Controller Design

Determination of definite bounds on the states of a stochastic process leads to an obvious application in controller design. The information from the Bernstein coefficients can now be used to satisfy state constraints in the design of a controller robust to the uncertainties in the model.

The controller is designed in a discrete time setting of the system. A LP formulation is used (as the constraints can be reduced to linear equalities and inequalities) to solve for the desired control input. The LP problem formulation for linear dynamical systems as elaborated by Singh [15] was discussed in Sec. 2. The extension of the LP approach, which incorporates the Bernstein bounds is discussed next.

7.1 Linear Programming Problem for Stochastic Processes.

The LP formulation for a stochastic process has a similar framework as that of the deterministic problem. However, the primary difference lies in the fact that even though the desired final states are known, the final state values for a stochastic process remains unknown.

To formulate the problem considering uncertainty, the residual energy of the final states is chosen to be the cost to be minimized. The residual potential energy is only a function of the position states and the residual kinetic energy is only a function of the velocity states. Hence, constraining the residual energy constrains the final state residues to become minimum.

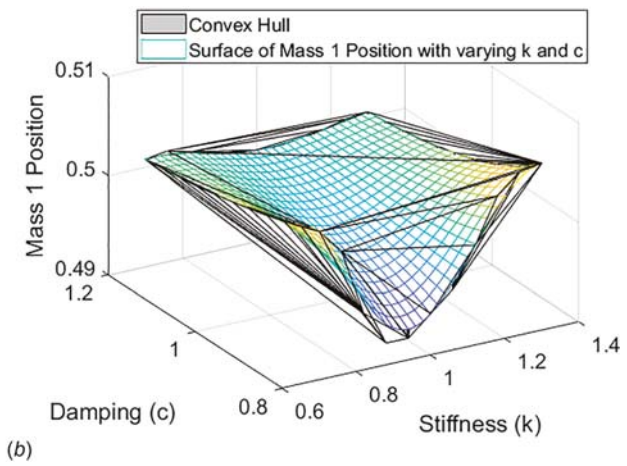
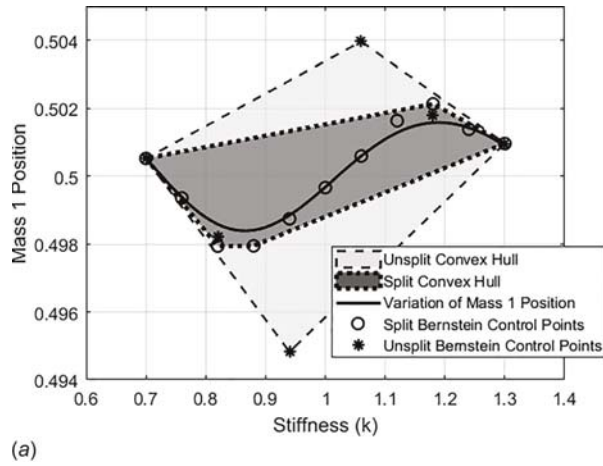


Fig. 7 Split Bernstein coefficients and split convex hulls: (a) split coefficients at time = 29 and hull (1D) and (b) split coefficients at time = 27 and hull (2D)

Table 2 1D split control points

k	b_{1i}	k	b_{1i}	k	b_{1i}
0.7000	0.5005	0.7600	0.4994	0.8200	0.4979
0.8800	0.4979	0.9400	0.4987	1.0000	0.4997
1.0000	0.4997	1.0600	0.5006	1.1200	0.5016
1.1800	0.5021	1.2400	0.5014	1.3000	0.5010

Table 3 2D unsplit control points

$c \backslash k$	0.7000	0.8200	0.9400	1.0600	1.1800	1.3000
0.8000	0.5050	0.5036	0.5025	0.5017	0.5012	0.5008
0.8800	0.5000	0.5021	0.5023	0.5022	0.5018	0.5015
0.9600	0.4779	0.4871	0.4921	0.4952	0.4971	0.4983
1.0400	0.4993	0.4974	0.4973	0.4975	0.4979	0.4983
1.1200	0.5061	0.5028	0.5014	0.5007	0.5003	0.5001
1.2000	0.5068	0.5038	0.5023	0.5014	0.5009	0.5005

However, the residual energy is a quadratic term and does not fit the requirements of an LP problem. Hence, the residual energy constraint is approximated to conform to LP rules.

For linear systems

$$M\ddot{z} + C\dot{z} + Kz = Du \quad (44)$$

where M , C , and K are mass, damping, and stiffness matrices and D is the control influence matrix; the energy (E) at any time instant is given by

$$E = \frac{1}{2} \dot{z}^T M \dot{z} + \frac{1}{2} z^T K z = \frac{1}{2} (\sqrt{M} \dot{z})^T (\sqrt{M} \dot{z}) + \frac{1}{2} (\sqrt{K} z)^T (\sqrt{K} z) \quad (45)$$

which is the sum of the kinetic and potential energies. It can be also visualized as the l_2 norm of the square root of potential and kinetic energies. To make the residual energy constraint compatible with a LP problem, a linear approximation of Eq. (45) is assumed. This is done by using an l_1 or a l_∞ norm instead of the l_2 norm (which is essentially circumscribing or inscribing the l_2 hypersphere with bounding hyperplanes).

Linear systems (i.e., Eq. (44)) can be written in state space as

$$\begin{Bmatrix} \dot{z} \\ z \end{Bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}}_{A'} \underbrace{\begin{Bmatrix} z \\ z \end{Bmatrix}}_Z + \underbrace{\begin{bmatrix} 0 \\ M^{-1}D \end{bmatrix}}_{B'} u \quad (46)$$

After discretization, the model is given as

$$Z(k+1) = AZ(k) + Bu(k) \quad (47)$$

The terminal time error states are defined as

$$X = \begin{Bmatrix} x(N_t+1) \\ \dot{x}(N_t+1) \end{Bmatrix} = \begin{Bmatrix} z(N_t+1) - z_f \\ \dot{z}(N_t+1) - \dot{z}_f \end{Bmatrix} \quad (48)$$

where z_f and $\dot{z}_f$ are the desired final states.

A new set of states (at final time) are defined (which will be used to implement the residual energy constraint later) as

Table 4 2D completely split control points

$c \backslash k$	0.7000	0.7600	0.8200	0.8800	0.9400	1.0000	1.0600	1.1200	1.1800	1.2400	1.3000
0.8000	0.5050	0.5043	0.5037	0.5032	0.5027	0.5023	0.5019	0.5015	0.5012	0.5010	0.5008
0.8400	0.5025	0.5027	0.5027	0.5025	0.5024	0.5022	0.5020	0.5017	0.5015	0.5013	0.5011
0.8800	0.4957	0.4972	0.4983	0.4990	0.4995	0.4998	0.5001	0.5003	0.5004	0.5005	0.5005
0.9200	0.4922	0.4942	0.4956	0.4966	0.4974	0.4981	0.4987	0.4991	0.4994	0.4996	0.4998
0.9600	0.4922	0.4938	0.4951	0.4961	0.4968	0.4974	0.4981	0.4985	0.4989	0.4991	0.4994
1.0000	0.4942	0.4952	0.4960	0.4967	0.4972	0.4977	0.4981	0.4985	0.4988	0.4990	0.4992
1.0400	0.4961	0.4965	0.4969	0.4973	0.4976	0.4979	0.4982	0.4985	0.4987	0.4989	0.4991
1.0800	0.5001	0.4995	0.4992	0.4990	0.4990	0.4990	0.4990	0.4990	0.4991	0.4992	0.4993
1.1200	0.5045	0.5031	0.5021	0.5015	0.5010	0.5006	0.5002	0.5000	0.4999	0.4998	0.4998
1.1600	0.5064	0.5049	0.5037	0.5029	0.5022	0.5017	0.5012	0.5009	0.5006	0.5005	0.5003
1.2000	0.5068	0.5053	0.5042	0.5033	0.5027	0.5021	0.5016	0.5012	0.5009	0.5007	0.5005

$$\begin{Bmatrix} \mathbf{y}(N_t + 1) \\ \dot{\mathbf{y}}(N_t + 1) \end{Bmatrix} = \begin{Bmatrix} \sqrt{K}\mathbf{x}(N_t + 1) \\ \sqrt{M}\dot{\mathbf{x}}(N_t + 1) \end{Bmatrix} \quad (49)$$

where $\mathbf{y} = [y_1, y_2, \dots, y_r]^T$, $\dot{\mathbf{y}} = [\dot{y}_1, \dot{y}_2, \dots, \dot{y}_r]^T$ and r is the dimension of $\mathbf{z}$.

7.1.1 l_∞ Formulation. The final states, and hence the residual states, are a linear function of the control input. The new set of states ($\mathbf{y}$ and $\dot{\mathbf{y}}$) defined by Eq. (49) are linear transformations of the residual states. Thus, a set of linear inequalities can be framed to solve for the minimax problem.

The final problem can be posed as

$$\begin{aligned} & \text{minimize}_{u,f} f \\ & \text{subject to} \quad -f \leq y_1^i(N_t + 1) \leq f \\ & \quad \quad \quad -f \leq y_2^i(N_t + 1) \leq f \\ & \quad \quad \quad \vdots \\ & \quad \quad \quad -f \leq y_r^i(N_t + 1) \leq f \quad \forall i = 0, 1, \dots, p \\ & \quad \quad \quad -f \leq \dot{y}_1^i(N_t + 1) \leq f \\ & \quad \quad \quad \vdots \\ & \quad \quad \quad -f \leq \dot{y}_r^i(N_t + 1) \leq f \\ & \quad \quad \quad u_{lb} \leq u(k) \leq u_{ub} \quad \forall k = 1, 2, \dots, N_t \\ & \quad \quad \quad \text{state constraints} \quad \forall k = 1, 2, \dots, N_t \end{aligned}$$

where p is the number of uncertain models selected over the space of uncertainty on which the residue is to be minimized, u_{lb} and u_{ub} are the bounds on the control and the State Constraints are the desired state restrictions during the control operation.

7.1.2 l_1 Formulation. The l_1 problem can be formulated as

$$\begin{aligned} & \text{minimize}_{u,f} f \\ & \text{subject to} \quad \left[\sum_{s=1}^r C_{s,y_s^i}(N_t + 1) \leq f \right] \quad \text{for } i = 1, 2, \dots, p \\ & \quad \quad \quad u_{lb} \leq u(k) \leq u_{ub} \quad \forall k = 1, 2, \dots, N_t \\ & \quad \quad \quad \text{state constraints} \quad \forall k = 1, 2, \dots, N_t \end{aligned}$$

where $C_s = \pm 1$; r is the dimension of $\mathbf{z}$, p is the number of models selected within the region of uncertainty, and k the time index of the discrete system. As each C_s can have two values (± 1), the total number of residual energy constraints becomes 2^r .

7.2 Linear Programming Problem With Bernstein Coefficients. In Secs. 7.1.1 and 7.1.2, it was stated that the number of models used to form the constraints was p . Accurate representations of multivariate stochastic systems with sampling require a large number of such samples. The magnitude of this number exponentially increases with the dimensionality of the uncertainty [32]. Hence, the robustness of the controller is limited by the number of samples of the model chosen.

However, when the stochastic process has a Bernstein formulation, the control points of the Bernstein expansion provide absolute deterministic bounds on the states. Hence, a smart sampling method, which requires the model samples to be the control points (which make the convex hull), always makes the control design valid.

The case for using the unsplit control points is first presented before the split control points case is presented. Once again, the same example is used to show the theory discussed.

7.2.1 Illustrative Example. In this section, the method to formulate the LP problem and solve it has been shown. Considering the model from Fig. 1, the dynamics are given by the matrices

$$M = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \quad (50)$$

For this example, it is once again assumed that the uncertainty in the model is one-dimensional. The uncertainty lies in the spring constant value given by Eq. (17).

To calculate the l_1 or l_∞ form of the residual energy, the square root of K matrix is desired. Thus, a pseudo spring (with spring constant k_p) is attached to the first mass to make the stiffness matrix positive definite, which is required when defining a residual energy cost function for the linear programming problem. Therefore

$$K_h = \begin{bmatrix} k + k_p & -k \\ -k & k \end{bmatrix} \quad (51)$$

For all simulations, K_h is used as the effective stiffness matrix with the value of k_p being 0.05.

As seen previously, in order to formulate the LP problem, a discrete setting is necessary. The whole procedure for PC expansion is followed to obtain the Legendre states in discrete format. After similar formulations to Eqs. (19)–(22), followed by a Galerkin projection, a set of deterministic equations are derived similar to Eqs. (23)–(25)

$$\dot{z}_{10}(\Psi_0, \Psi_0) = f_{10}(z_{10}, \dots, z_{45}, u) \quad (52)$$

$\vdots$

$$\dot{z}_{45}(\Psi_5, \Psi_5) = f_{45}(z_{10}, \dots, z_{45}, u) \quad (53)$$

Equations (52) and (53) are discretized to yield

$$\mathbf{Z}(k+1) = \mathbf{A}\mathbf{Z}(k) + \mathbf{B}u(k) = \mathbf{A}^k\mathbf{Z}(1) + \sum_{i=1}^k \mathbf{A}^{k-i}\mathbf{B}u(i) \quad (54)$$

where $\mathbf{Z} = [z_{10}, \dots, z_{45}]^T$ are the Legendre coefficients. Therefore

$$\mathbf{Z}(T_f) = \mathbf{Z}(N_t + 1) = \mathbf{A}^{N_t}\mathbf{Z}(1) + \sum_{i=1}^{N_t} \mathbf{A}^{N_t-i}\mathbf{B}u(i) \quad (55)$$

The transformation from Legendre states to Bernstein states is linear and the transformation matrix is said to be represented by M_N . Therefore, the terminal states in Bernstein form are

$$\mathbf{Z}_B(T_f) = M_N\mathbf{Z}(N_t + 1) = M_N\mathbf{A}^{N_t}\mathbf{Z}(1) + \sum_{i=1}^{N_t} M_N\mathbf{A}^{N_t-i}\mathbf{B}u(i) \quad (56)$$

$$\text{or } \mathbf{Z}_B(T_f) = M_N\mathbf{A}^{N_t}\mathbf{Z}(1) + [M_N\mathbf{A}^{N_t-1}\mathbf{B} \dots M_N\mathbf{B}] \begin{bmatrix} u(1) \\ \vdots \\ u(N_t) \end{bmatrix} \quad (57)$$

where $\mathbf{Z}_B$ is a vector of dimension $(n(N+1) = 24)$ containing all the Bernstein coefficients of the Bernstein basis functions obtained under the influence of the input $u(k)$. The objective of the example is to find a control input, which can drive the system from its initial states $([0, 0, 0, 0]^T)$ to its final states $([1, 1, 0, 0]^T)$ while constraining the input to remain within the domain $[-1, 1]$ and constraining the relative displacement of the masses to remain within 0.2. From the desired final states, the desired Bernstein coefficients at the final time can be easily determined as

$$\mathbf{Z}_{Bd}(T_f) = [1, \dots, (12), \dots, 1, 0, \dots, (12), \dots, 0]^T \quad (58)$$

Thus, the residual states are

$$X(T_f) = \begin{bmatrix} \mathbf{x}_1(T_f) \\ \mathbf{x}_2(T_f) \\ \dot{\mathbf{x}}_1(T_f) \\ \dot{\mathbf{x}}_2(T_f) \end{bmatrix} = \mathbf{Z}_B(T_f) - \mathbf{Z}_{Bd}(T_f) \quad (59)$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors of dimension 6 having the Bernstein coefficients of the final residual positions of mass 1 and 2, respectively. Similarly, $\dot{\mathbf{x}}_1$ and $\dot{\mathbf{x}}_2$ hold the coefficients for their residual velocities.

Now, depending on the way in which the residual energy constraint is implemented (l_1 or l_∞), separate formulations of the LP problem can be made.

Defining new states similar to Eq. (49)

$$y_1^{(i)} = \sqrt{K_{h(1,1)}^{(i)}} \mathbf{x}_1^{(i)} + \sqrt{K_{h(1,2)}^{(i)}} \mathbf{x}_2^{(i)} \quad (60)$$

$$y_2^{(i)} = \sqrt{K_{h(2,1)}^{(i)}} \mathbf{x}_1^{(i)} + \sqrt{K_{h(2,2)}^{(i)}} \mathbf{x}_2^{(i)} \quad (61)$$

$$y_3^{(i)} = \sqrt{M_{(1,1)}} \dot{\mathbf{x}}_1^{(i)} + \sqrt{M_{(1,2)}} \dot{\mathbf{x}}_2^{(i)} \quad (62)$$

$$y_4^{(i)} = \sqrt{M_{(2,1)}} \dot{\mathbf{x}}_1^{(i)} + \sqrt{M_{(2,2)}} \dot{\mathbf{x}}_2^{(i)} \quad (63)$$

where $i = 0, 1, \dots, p$ are the sample models and $\mathbf{x}_a^{(i)}$ is the i th member of $\mathbf{x}_a$. In this framework, as stated earlier, the samples are selected to be the control points of the convex hull enveloping the uncertain region. Therefore, $p = N = 5$. $\sqrt{K_{h(r,c)}^{(i)}}$ refers to the r th row and c th column of the matrix $\sqrt{K_h}$ defined for $k^{(i)}$ where $k^{(i)}$ is defined as

$$k^{(i)} = 0.7 + (1.3 - 0.7) \frac{i}{N} \quad (64)$$

l_∞ Formulation. For an l_∞ formulation, the nature of constraints is provided in Sec. 7.1.1. Since there are four states in the model, the constraint equations are

$$-f \leq y_1^{(i)} \leq f; \quad -f \leq y_2^{(i)} \leq f \quad (65)$$

$$-f \leq y_3^{(i)} \leq f; \quad -f \leq y_4^{(i)} \leq f \quad (66)$$

These constraints can be exercised with the help of an output matrix C_{l_∞} . The constraints can therefore be written in a matrix format as

$$\underbrace{\begin{bmatrix} \pm [C_{l_\infty} M_N A^{N_i-1} B & \dots & C_{l_\infty} M_N B & \mp I_{24,1}] \\ H_\pm \end{bmatrix}}_{H_\pm} \underbrace{\begin{bmatrix} u(1) \\ \vdots \\ u(N_t) \\ f \end{bmatrix}}_{\tilde{\mathbf{z}}} \leq \pm \mathbf{h}_1 \quad (67)$$

where $\mathbf{h}_1$ is the vector of constant terms derived from Eqs. (65) and (66). Hence, the final problem can be stated as

$$\begin{aligned} &\text{minimize} && f = \tilde{\mathbf{c}}^T \tilde{\mathbf{z}} \\ &\text{subject to} && H \tilde{\mathbf{z}} \leq \mathbf{h} \\ &&& \tilde{\mathbf{z}}_{\min} \leq \tilde{\mathbf{z}} \leq \tilde{\mathbf{z}}_{\max} \\ &&& |\text{relative mass position}| \leq 0.2 \end{aligned} \quad (68)$$

where

$$\tilde{\mathbf{c}}^T = [0 \dots 0 \ 1], \quad \tilde{\mathbf{z}} = [u(1) \dots u(N_t) f]^T \quad (69)$$

$$H = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ -\mathbf{h}_1 \end{bmatrix} \quad (70)$$

$$\begin{bmatrix} u_{\min}(1) \\ \vdots \\ u_{\min}(N_t) \\ 0 \end{bmatrix} \leq \tilde{\mathbf{z}} \leq \begin{bmatrix} u_{\max}(1) \\ \vdots \\ u_{\max}(N_t) \\ \infty \end{bmatrix} \quad (71)$$

l_1 Formulation. Constraints for an l_1 norm formulation (similar to Sec. 7.1.2) can be stated as

$$-f \leq y_1^{(i)} \pm y_2^{(i)} \pm y_3^{(i)} \pm y_4^{(i)} \leq f \quad (72)$$

Note that Eq. (72) is a shorthand for eight constraints. Once again, these constraints can be expressed through an output matrix C_{l_1} . The final problem can be posed in a similar fashion as the l_∞ norm formulation described in Sec. 7.1.2.

On solving these LP problems, the resulting residual energy as a function of the uncertain stiffness for both the l_1 and l_∞ can be calculated to illustrate the relative robustness of the solutions. Figure 8(a) shows the variation of residual energy with stiffness for both norms and the worst cost is comparable. Figures 8(b) and 8(c) show the residual state distributions over the entire region of uncertainty.

7.3 Linear Programming Problem With Split Bernstein Coefficients. Using the previously illustrated example, the methodology to apply the split Bernstein coefficients to the LP problem is shown in this section.

Initially, the order of PC expansion chosen was ($N = 5$). Hence, each state was represented by ($N + 1 = 6$) coefficients. Since there are four states in our chosen problem, the total number of coefficients were $((N + 1) \times 4 = 24)$. As the Bernstein coefficients are split, the Bernstein coefficients representing each state increases to $(2N + 1 = 11)$ and the total number of coefficients become $(11 \times 4 = 44)$.

Following the procedure described previously, the desired set of Bernstein coefficients at the final time (T_f) are now given by:

$$\mathbf{Z}_{Bd}(T_f) = [1, \dots (22) \dots, 1, 0, \dots (22) \dots, 0]^T \quad (73)$$

where $\mathbf{Z}_{Bd}(T_f) \in \mathbb{R}^{44}$.

Consider a matrix $M_B(a)$, which when multiplied with a vector containing the unsplit Bernstein coefficients of all the states, gives a vector containing all the split Bernstein coefficients (split at the point a)

$$\mathbf{B}_s = M_B(a) \times \mathbf{B}_{us} \quad (74)$$

where $\mathbf{B}_s$ and $\mathbf{B}_{us}$ are split and unsplit coefficients of the same states, respectively. As the initial PC order was $N = 5$ in the problem, $M_B \in \mathbb{R}^{4(2N+1) \times 4(N+1)}$ and is used to enforce the constraints previously discussed.

The terminal states in unsplit Bernstein form (as given by Eq. (56)) can now be rewritten in split Bernstein form as

$$\mathbf{Z}_B(T_f) = M_B M_N A^{N_t} \mathbf{Z}(1) + [M_B M_N A^{N_t-1} B \dots M_B M_N B] \begin{bmatrix} u(1) \\ \vdots \\ u(N_t) \end{bmatrix} \quad (75)$$

where $\mathbf{Z}_B(T_f) \in \mathbb{R}^{44}$.

The residual states, thus, become

$$\begin{aligned} \mathbf{X}(T_f) &= [\mathbf{x}_1(T_f) \quad \mathbf{x}_2(T_f) \quad \dot{\mathbf{x}}_1(T_f) \quad \dot{\mathbf{x}}_2(T_f)]^T \\ &= \mathbf{Z}_B(T_f) - \mathbf{Z}_{Bd}(T_f) \end{aligned} \quad (76)$$

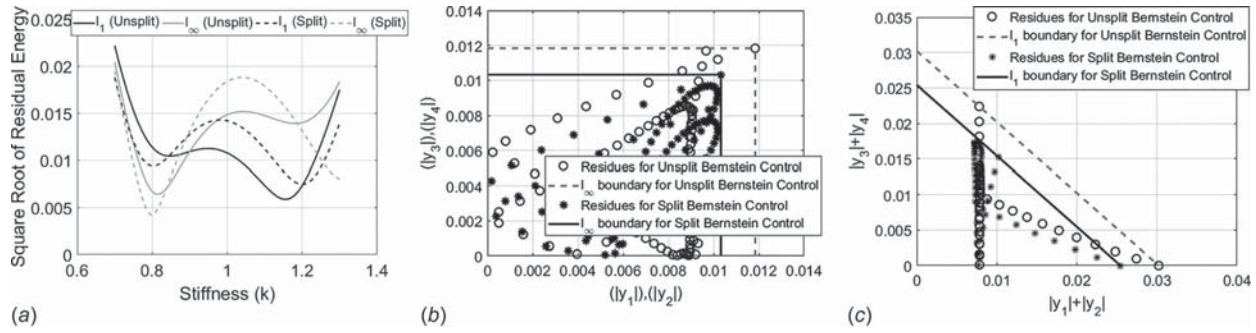


Fig. 8 Residual sensitivities and state distributions for unsplit bounds: (a) residual energy sensitivity, (b) residual state distribution: l_∞ form, and (c) residual state distribution: l_1 form

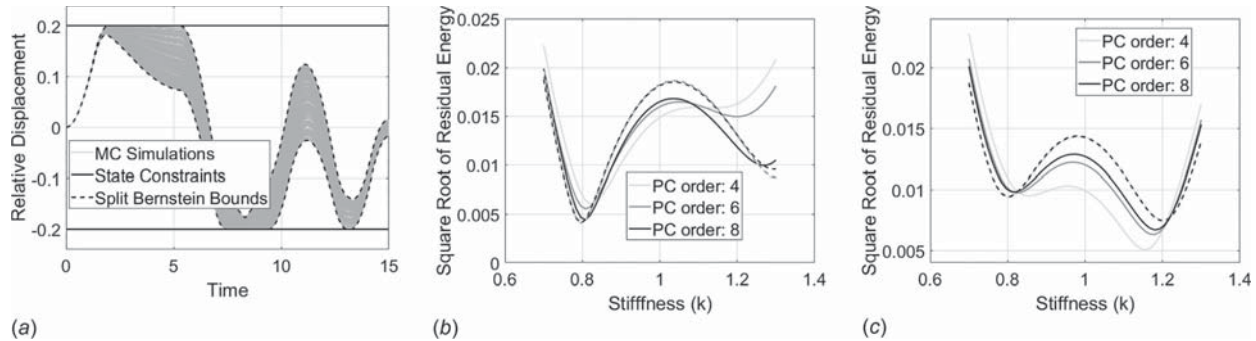


Fig. 9 (a) MC simulations showing all trajectories meet the state constraints, (b) and (c) comparison of residual energy sensitivities: (a) relative displacement with Bernstein envelope, (b) l_∞ formulation, and (c) l_1 formulation

where x_1 and x_2 are vectors of dimension 11 having the Bernstein coefficients of the final residual positions of mass 1 and 2, respectively. Similarly, $\dot{x}_1$ and $\dot{x}_2$ hold the coefficients for their velocities.

New states are once again defined by Eqs. (60)–(63). As the number of control points have increased on splitting the Bernstein coefficients, the number of samples increase to $p = 2N + 1 = 11$. All the Bezier curves (i.e., the residual state curves with respect to stiffness) were split at their midpoints (i.e., at $k = 1$ or $\xi = 0.5$). Hence, $k^{(i)}$ for this scheme of implementation is defined as

$$k^{(i)} = 0.7 + (1.3 - 0.7) \frac{i}{2N + 1} \quad \forall i = 0, 1, \dots, p = 11 \quad (77)$$

The rest of the formulation remains almost identical. Considering C_l as the output matrix (depending on the l_1 or the l_∞ formulation), the inequality constraints are given by

$$\underbrace{\pm [C_l M_B M_N A^{N_i-1} B \dots C_l M_B M_N B, \mp 1]}_{H_\pm} \underbrace{\begin{bmatrix} u(1) \\ \vdots \\ u(N_t) \\ f \end{bmatrix}}_z \leq \pm h_1 \quad (78)$$

where h_1 is the vector of constant terms derived from the constraint equations. The final problem can be summarized by Eqs. (68)–(71).

On solving the split LP problem, the control (obtained from l_∞) is used to make MC simulations of the relative displacement of the masses (Fig. 9(a)) and is shown to lie within the constraints (≤ 0.2) as well as within the envelope of Bernstein bounds.

Figure 8(a) shows the residual energy sensitivity, and Figs. 8(b) and 8(c) show the residual state distributions. Comparisons with the unsplit Bernstein control show that the split residual energy is consistently lower than the unsplit counterpart.

Figures 9(b) and 9(c) show a comparison between the unsplit and split methods based on residual energy sensitivity. The dotted curves correspond to the split method while the solid lines correspond to the unsplit method. The difference in color corresponds to results obtained from different orders of PC expansion.

Once again, it is evident that in the case of the split method, the residual energy has lower values at the edges (i.e., worst case values). Also, as expected, with an increase in the order of the PC expansion, the worst case residual energy improves.

8 Conclusion

This paper proposes to exploit the Bernstein Polynomials to determine the bounds of states of uncertain dynamical systems. A two mass spring system is used to illustrate the development of a set of deterministic differential equations after parameterizing the time-invariant model parameter uncertainties using polynomial chaos series. The coefficient of the Bernstein polynomials defines a convex hull, which permits determination of the upper and lower bounds of the uncertain evolution of the system states. The splitting of the Bernstein coefficients allows the development of a tighter convex hull. Finally, a rest-to-rest maneuver for a system with uncertain stiffness is used to illustrate the design of a state constrained controller, which is robust to model parameter uncertainties.

Funding Data

- National Science Foundation, Directorate for Engineering (CMMI 1537210).

References

- [1] Singh, T., and Vadali, S. R., 1993, "Robust Time-Delay Control," *ASME J. Dyn. Syst. Meas. Control*, **115**(2), pp. 303–306.
- [2] Singer, N. C., and Seering, W. P., 1990, "Preshaping Command Inputs to Reduce System Vibrations," *ASME J. Dyn. Syst. Meas. Control*, **112**(1), pp. 76–82.

- [3] Singh, T., and Vadali, S. R., 1994, "Robust Time-Optimal Control: Frequency Domain Approach," *J. Guid. Control Dyn.*, **17**(2), pp. 346–353.
- [4] Singh, T., 2002, "Minimax Design of Robust Controllers for Flexible Structures," *J. Guid. Control Dyn.*, **25**(5), pp. 868–875.
- [5] Liu, Q., and Wie, B., 1992, "Robust Time-Optimal Control of Uncertain Flexible Spacecraft," *J. Guid. Control Dyn.*, **15**(3), pp. 597–604.
- [6] Liu, S., Garimella, P., and Yao, B., 2005, "Adaptive Robust Precision Motion Control of a Flexible System With Unmatched Model Uncertainties," American Control Conference (ACC), Portland, OR, June 8–10, pp. 750–755.
- [7] Singhose, W., Banerjee, A. K., and Seering, W. P., 1997, "Slewing Flexible Spacecraft With Deflection Limiting Input Shaping," *AIAA J. Guid. Control Dyn.*, **20**(2), pp. 291–298.
- [8] Vossler, M., and Singh, T., 2010, "Minimax State/Residual-Energy Constrained Shapers for Flexible Structures: Linear Programming Approach," *ASME J. Dyn. Syst., Meas. Control*, **132**(3), p. 031010.
- [9] Bhattacharya, R., 2015, "A Polynomial Chaos Framework for Designing Linear Parameter Varying Control Systems," American Control Conference (ACC), Chicago, IL, July 1–3, pp. 409–414.
- [10] Kim, K.-K., and Braatz, R., 2012, "Generalized Polynomial Chaos Expansion Approaches to Approximate Stochastic Receding Horizon Control With Applications to Probabilistic Collision Checking and Avoidance," IEEE Conference on Control Applications (CCA), Dubrovnik, Croatia, Oct. 3–5, pp. 350–355.
- [11] Singh, T., Singla, P., and Konda, U., 2010, "Polynomial Chaos Based Design of Robust Input Shapers," *ASME J. Dyn. Syst. Meas. Control*, **132**(5), p. 051010.
- [12] Terejanu, G., Singla, P., Singh, T., and Scott, P., 2010, "Approximate Propagation of Both Epistemic and Aleatory Uncertainty Through Dynamic Systems," 13th International Conference on Information Fusion (ICIF), Edinburgh, UK, July 26–29, pp. 1–8.
- [13] Monti, A., Ponci, F., and Lovett, T., 2005, "A Polynomial Chaos Theory Approach to Uncertainty in Electrical Engineering," 13th International Conference on Intelligent Systems Application to Power Systems (ISAP), Arlington, VA, Nov. 6–10, pp. 7–10.
- [14] Nandi, S., Migeon, V., Singh, T., and Singla, P., 2016, "State Constrained Controller Design for Uncertain Linear Systems Using Polynomial Chaos," American Control Conference (ACC), Boston, MA, July 6–8, pp. 2005–2010.
- [15] Singh, T., 2009, *Optimal Reference Shaping for Dynamical Systems: Theory and Applications*, CRC Press, Boca Raton, FL.
- [16] Singh, T., 2008, "Minimax Input Shaper Design Using Linear Programming," *ASME J. Dyn. Syst. Meas. Control*, **130**(5), p. 051010.
- [17] Wiener, N., 1938, "The Homogeneous Chaos," *Am. J. Math.*, **60**(4), pp. 897–936.
- [18] Cameron, R. H., and Martin, W. T., 1947, "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals," *Ann. Math.*, **48**(2), pp. 385–392.
- [19] Ghanem, R. G., and Spanos, P. D., 1991, *Stochastic Finite Elements: Aspectral Approach*, Springer, New York.
- [20] Xiu, D., and Karniadakis, G. E., 2002, "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM J. Sci. Comput.*, **24**(2), pp. 619–644.
- [21] Garloff, J., 2003, "The Bernstein Expansion and Its Applications," *J. Am. Roum. Acad.*, **25**(27), pp. 80–85.
- [22] Ray, S., and Nataraj, P., 2009, "An Efficient Algorithm for Range Computation of Polynomials Using the Bernstein Form," *J. Global Optim.*, **45**(3), pp. 403–426.
- [23] Garloff, J., 1993, "The Bernstein Algorithm," *Interval Comput.*, **2**(6), pp. 154–168.
- [24] Berg, M. D., Cheong, O., Kreveld, M. V., and Overmars, M., 2008, *Computational Geometry: Algorithms and Applications*, 3rd ed., Springer-Verlag TELOS, Santa Clara, CA.
- [25] Farouki, R., 2000, "Legendre-Bernstein Basis Transformations," *J. Comput. Appl. Math.*, **119**(1–2), pp. 145–160.
- [26] Rababah, A., 2004, "Jacobi-Bernstein Basis Transformation," *Comput. Methods Appl. Math.*, **4**(2), pp. 206–214.
- [27] Moore, R. E., 1966, *Interval Analysis*, Vol. 4, Prentice Hall, Englewood Cliffs, NJ.
- [28] Althoff, M., Stursberg, O., and Buss, M., 2007, "Reachability Analysis of Linear Systems With Uncertain Parameters and Inputs," 46th IEEE Conference on Decision and Control (CDC), New Orleans, LA, Dec. 12–14, pp. 726–732.
- [29] Goldsztejn, A., 2009, "On the Exponentiation of Interval Matrices," preprint [arXiv:0908.3954](https://arxiv.org/abs/0908.3954).
- [30] Stolfi, J., and De Figueiredo, L., 2003, "An Introduction to Affine Arithmetic," *Trends Appl. Comput. Math.*, **4**(3), pp. 297–312.
- [31] De Casteljau, P., 1959, "Outils méthodes calcul," *André Citroën Automobiles SA*, Paris, France.
- [32] Stroud, A. H., and Secrest, D., 1966, *Gaussian Quadrature Formulas*, Vol. 39, Prentice Hall, Englewood Cliffs, NJ.