Motion Planning of Uncertain Ordinary Differential Equation Systems

Joe Hays
Control Systems Branch,
Spacecraft Engineering Division,
Naval Center for Space Technology,
US Naval Research Laboratory,
Washington, DC 20375
e-mail: joehays@vt.edu

Adrian Sandu
Computational Science Laboratory,
Computer Science Department,
Virginia Tech,
Blacksburg, VA 24061
e-mail: sandu@cs.vt.edu

Corina Sandu
Advanced Vehicle Dynamics Laboratory,
Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24061
e-mail: csandu@vt.edu

Dennis Hong
Robotics and Mechanisms Laboratory,
Mechanical Engineering,
Virginia Tech,
Blacksburg, VA 24061
e-mail: dhong@vt.edu

This work presents a novel motion planning framework, rooted in nonlinear programming theory, that treats uncertain fully and underactuated dynamical systems described by ordinary differential equations. Uncertainty in multibody dynamical systems comes from various sources, such as system parameters, initial conditions, sensor and actuator noise, and external forcing. Treatment of uncertainty in design is of paramount practical importance because all real-life systems are affected by it, and poor robustness and suboptimal performance result if it is not accounted for in a given design. In this work uncertainties are modeled using generalized polynomial chaos and are solved quantitatively using a least-square collocation method. The computational efficiency of this approach enables the inclusion of uncertainty statistics in the nonlinear programming optimization process. As such, the proposed framework allows the user to pose, and answer, new design questions related to uncertain dynamical systems. Specifically, the new framework is explained in the context of forward, inverse, and hybrid dynamics formulations. The forward dynamics formulation, applicable to both fully and underactuated systems, prescribes deterministic actuator inputs that yield uncertain state trajectories. The inverse dynamics formulation is the dual to that of forward dynamics, and is only applicable to fully actuated systems; deterministic state trajectories are prescribed and yield uncertain actuator inputs. The inverse dynamics formulation is more computationally efficient as it requires only algebraic evaluations and completely avoids numerical integration. Finally, the hybrid dynamics formulation is applicable to underactuated systems where it leverages the benefits of inverse dynamics for actuated joints and forward dynamics for unactuated joints; it prescribes actuated state and unactuated input trajectories that yield uncertain unactuated states and uncertain actuated inputs. The benefits of the ability to quantify uncertainty when planning the motion of multibody dynamic systems are illustrated through several case studies. The resulting designs determine optimal motion plans—subject to deterministic and statistical constraints—for all possible systems within the probability space. [DOI: 10.1115/1.4026994]

1 Introduction

1.1 Motivation. Design engineers cannot quantify exactly every aspect of a given system. These uncertainties frequently create difficulties in accomplishing design goals and can lead to poor robustness and suboptimal performance. Tools that facilitate the analysis and characterization of the effects of uncertainties enable designers to develop more robustly performing systems. The need to analyze the effects of uncertainty is particularly acute when designing motion plans for dynamical systems. Frequently, engineers do not account for various uncertainties in their motion plan in order to save time and to reduce costs. However, this simply delays, or hides, the cost, which is inevitably incurred downstream in the design flow, or worse, after the system has been deployed and fails to meet the design goals. Ultimately, if a robust motion plan is to be achieved, uncertainties must be accounted for up-front during the design process.

Many industries employ dynamic systems with planned motions that operate with uncertainty. For example, the industrial manufacturing sector uses articulated robotic systems for repeated tasks such as welding, packaging, and assembly; medical robots have been designed to aid physicians in surgery; and autonomous vehicles are taking on more and more tasks in military, municipality, and even domestic operations.

In the area of unmanned ground vehicles (UGVs), organizations such as the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation (NSF), Office of Naval Research (ONR), and other agencies continue to investigate the application of legged robotic systems. Additionally, many UGVs, unmanned surface vehicles (USVs), and unmanned underwater vehicles (UUVs) are outfitted with articulated accessories to perform various tasks. These systems are designed to aid in diverse operations including improvised incendiary device (IID) detection and disarmament, material and equipment handling, and convoy, search, and rescue. Some showcase examples include Boston dynamics' BigDog (and next generation LS3), which convoys soldier equipment and Vecna's BEAR that retrieves wounded soldiers. Both of these examples work with payloads of varying size and weight in rough unknown terrain. These examples clearly illustrate the need to design motion strategies with uncertainties in mind. Elaborating further on the equipment convoy task, optimal design of the locomotion strategy, or gait, of the systems carrying uncertain payloads could result in large fuel/energy savings as well as lengthen the achievable distances of a given operation.

1.2 State of the Art in Motion Planning and Uncertainty Quantification. In the following, a review of the literature is presented where works related to motion planning and uncertainty quantification are specifically covered.

1.2.1 Deterministic Optimization-Based Motion Planning. In Ref. [1] Park presents a nonlinear programming approach to motion planning for robotic manipulator arms described by

Contributed by the Design Engineering Division of ASME for publication in the Journal of Computational and Nonlinear Dynamics. Manuscript received March 14, 2012; final manuscript received March 1, 2014; published online May 13, 2014. Assoc. Editor: Jozsef Kovecses.

deterministic ODEs. The main contribution of Park's work is to define new cost terms that capture actuator force limiting characteristics, where actuator velocities and resulting feasible torques are defined. Park's formulation utilizes quintic B-splines to provide a tractable finite-dimensional search space along with quasi-Newton-based solver methods (e.g., BFGS). Additionally, he approaches obstacle avoidance by defining distance constraints with the *growth function* technique from Ref. [2].

Sohl et al. presented a series of papers that document their excellent work in the area of optimal manipulator motion planning [3–5]. Their approach provides a few critical properties that streamline the problem. First, their geometric dynamics formulation has an equivalent recursive formulation that provides $O(n)$ computational complexity. Second, the use of the product-of-exponentials (POE) in their formulation provides a straightforward and new approach to calculating the analytic gradient of the optimal motion planning objective function (which subsequently improves performance in the optimization search). In Ref. [6] Martin and Bobrow present a minimum effort formulation for open chain manipulators. In Ref. [7] Sohl and Bobrow extend the work to address branched kinematic chains; in Refs. [8–10] they again extend the work to address underactuated manipulators; and in Refs. [11,12] the methods are applied to the specific design problem of maximizing the weightlifting capabilities of a Puma 762 Robot. Throughout this series of work the sequential quadratic programming (SQP) technique is used to solve the constrained optimization problem, however, in Ref. [13] a Newton type optimization algorithm is developed that reuses the analytic gradient and Hessian information from the geometric dynamics. In Ref. [14] Bobrow et al., further extend the work to solve infinite-dimensional problems using a sequence of linear–quadratic optimal control subproblems. Finally, Ref. [15] extended the geometric-based optimization methods to more general dynamic systems including those with closed-kinematic loops and redundant actuators and sensors.

Another inspiring body of research comes from Xiang et al. [16–23] where analytic derivatives for the optimization cost of general open, branched, and closed looped systems, described by recursive Lagrangian dynamics, is presented. Formulations are based on the Denavit–Hartenberg kinematic methods, cubic B-splines, and SQP-based solvers. Application emphasis focuses on the motion planning of overactuated 3D human figures, where models with as many as 23 DOFs and 54 actuators are used to design natural cyclic walking gaits. A combination of *inverse* and *forward dynamics* formulations are used, however, their formulation avoids explicit numerical integration (required in a *sequential nonlinear programming* (SeqNLP) methodology). Instead, their formulation makes use of the *simultaneous nonlinear programming* (SimNLP) methodology, which discretizes the EOMs over the trajectory of the system and treats the complete set of equations as equality constraints for the NLP. Therefore, the SimNLP has a much larger set of constraints than the SeqNLP approach but enjoys a more structured NLP that typically experiences faster convergence. (Note the definitions of SimNLP and SeqNLP come from Refs. [24,25].)

1.2.2 Motion Planning of Uncertain Systems. Very little research has been performed in the area motion planning of uncertain dynamical systems. LaValle treats sensor uncertainty with RRTs in Ref. [26]. Barraquand addresses both actuator and sensor uncertainty in a stochastic dynamic programming (DP) framework but this work only addresses the kinematics of the system [27]. Park also presents a kinematic-only motion planning solution for systems with sensor and actuator uncertainties based on the Fokker–Planck equation [28]. Erdmann's early work on the *back-projection* method also only addressed sensor and actuator noise and was limited to first-order dynamic models [29].

In Ref. [30] Kewlani presents an RRT planner for mobility of robotic systems based on gPC but refers to it as a stochastic response surface method (SRSM). This technique is similar in spirit to the work presented in this paper; however, the main difference is that Kewlani's solution is developed only for determining a feasible motion plan (given the use of the RRT technique). In Refs. [31–33] Hays et al. presented initial investigations of the framework presented in this paper, where the goal of the new framework is to provide an optimal motion plan for uncertain dynamical systems versus a feasible one.

1.2.3 Generalized Polynomial Chaos (gPC) Uncertainty Quantification. Generalized polynomial chaos (gPC) is a relatively new method that is rapidly being accepted in diverse applications. Its origins come from early work by Wiener in the 1930s where he introduced the idea of *homogenous chaoses* [34]. His work made use of Gaussian distributions and the Hermite orthogonal polynomials. Xiu and Karniadakis generalized the concept by expanding the list of supported probability distributions and associated orthogonal polynomials [35,36], where the Galerkin projection method (GPM) was initially used. In Refs. [36–38] Xiu showed an initial collocation method based on Lagrange interpolation. A number of methods for selecting the collocation point locations were presented including tensor products and Smolyak sparse grids.

In Ref. [39] Sandu et al. introduced the least-squares collocation method (LSCM) and used the roots of the associated orthogonal polynomials in selecting the sampling points. Cheng and Sandu showed that the LSCM maintains the exponential convergence of GPM yet was superior in computational speed in Ref. [40]. The Hammersley *low-discrepancy sequences* (LDS) data set was their preferred method in selecting collocation points and they presented a modified time stepping mechanism for solving stiff systems where an approximate Jacobian was used. The main benefit of quantifying uncertainties with gPC through LSCM over GPM is that the deterministic constitutive relations need not be modified to produce an uncertain set of constitutive equations. The deterministic equations are simply evaluated at appropriately selected sample points from the probability space. This drastically simplifies the burden of implementing uncertainty quantification on the practitioner.

1.2.4 Multielement gPC. The accuracy of gPC deteriorates over time in long simulations and is dependent on the continuity of the system. In an effort to address these two concerns, Wan and Karniadakis developed multielement gPC (MEgPC) [41,42]. This method discretizes the probability space into nonoverlapping partitions. Within each partition the traditional single element gPC is performed. Summing element integrations provides a complete integration of the full probability space. The algorithm presented adaptively partitioned the space based on estimates of error convergence. When an error estimate deteriorated to a specified point the element was split. The initial work was developed for the GPM methodology using uniform distributions. MEgPC was subsequently extended to arbitrary distributions in Refs. [43,44]. Foo developed a collocation-based MEgPC in Ref. [45] and further extended the method to support higher dimensions using ANOVA methods in Ref. [46].

As an alternative to MEgPC, Witteveen and Iaccarino developed a similar multielement method based on gPC called the simplex elements stochastic collocation (SESC) method. This method adaptively partitions the probability space using simplex elements coupled with Newton–Cotes quadrature. Their method has shown an $O(n)$ convergence as long as the approximating polynomial order is increased with the number of uncertainties.

1.2.5 Recent Applications of gPC/MEgPC. The origins of gPC come from thermal/fluid applications; however, its adoption in other areas continues to expand. Sandu et al. introduced its application to multibody dynamical systems in [39,40,47–51]. Significant work has been done applying it as a foundational element in parameter [35–38,52–70] and state estimation [71,72], as well as system identification [73]. Relatively recent work has applied gPC to both classical and optimal control system design [52,74,75], power systems [76], and mobile robots [77].

1.3 Contributions of This Work. This work presents a novel nonlinear programming (NLP) based motion planning framework that treats smooth, lumped-parameter, uncertain, and fully and underactuated dynamical systems described by ordinary differential equations (ODEs). Uncertainty in multibody dynamical systems comes from various sources, such as system parameters, initial conditions, sensor and actuator noise, and external forcing. Treatment of uncertainty in design is of paramount practical importance because all real-life systems are affected by it, and poor robustness and suboptimal performance results if it is not accounted for in a given design. System uncertainties are modeled using generalized polynomial chaos (gPC) and are solved quantitatively using a least-square collocation method (LSCM). The computational efficiencies of this approach enable the inclusion of uncertainty statistics in the NLP optimization process. As such, new design questions related to uncertain dynamical systems can now be answered through the new framework.

Specifically, this work presents the new framework for forward, inverse, and hybrid dynamics formulations. The forward dynamics formulation—applicable to both fully and underactuated systems—has prescribed deterministic actuator inputs and yields uncertain state trajectories. The inverse dynamics formulation is the dual to the forward dynamics formulation and is only applicable to fully actuated systems. It has prescribed deterministic state trajectories and yields uncertain actuator inputs. The inverse dynamics formulation is more computationally efficient as it is only an algebraic evaluation and completely avoids any numerical integration. Finally, the hybrid dynamics formulation is applicable to underactuated systems. It leverages the benefits of inverse dynamics for the actuated joints and forward dynamics for the unactuated joints. It has prescribed actuated state and unactuated input trajectories, which are deterministic, and yields uncertain unactuated states and uncertain actuated inputs. The benefits of the ability to quantify uncertainty when planning motion of multibody dynamic systems are illustrated in various optimal motion planning case studies. The resulting designs determine optimal motion plans—subject to deterministic and statistical constraints—for all possible systems within the probability space.

It is important to point out that the new framework is not dependent on the specific formulation of the dynamical equations of motion (EOMs); formulations such as Newtonian, Lagrangian, Hamiltonian, and geometric methodologies are all applicable. This work applies the analytical Lagrangian EOM formulation.

The structure of this paper is as follows. A brief review of Lagrangian dynamics is presented in Sec. 2. Section 3 discusses the well-studied motion planning problem for deterministic systems. Section 4 reviews the gPC methodology for uncertainty quantification. Section 5 introduces the new framework for motion planning of uncertain fully and underactuated dynamical systems based on the uncertain forward, inverse, and hybrid dynamics formulations with illustrating case studies. Concluding remarks are presented in Sec. 6.

2 Multibody Dynamics

The new framework presented in this work is not dependent on a specific EOM formulation. Formulations such as Newtonian, Lagrangian, Hamiltonian, and geometric methodologies are all applicable. This work applies the analytical Lagrangian EOM formulation. As a very brief overview, the Euler–Lagrange ODE formulation for a multibody dynamical system can be described by [78,79]

$$\begin{aligned} &\boldsymbol{M}(\boldsymbol{q}(t),\boldsymbol{\theta}(t))\dot{\boldsymbol{v}}(t)+\boldsymbol{C}(\boldsymbol{q}(t),\boldsymbol{v}(t),\boldsymbol{\theta}(t))\boldsymbol{v}(t)+\boldsymbol{N}(\boldsymbol{q}(t),\boldsymbol{v}(t),\boldsymbol{\theta}(t)) \\ &\quad=\boldsymbol{\mathcal{F}}(\boldsymbol{q}(t),\boldsymbol{v}(t),\dot{\boldsymbol{v}}(t),\boldsymbol{\theta}(t))=\boldsymbol{S}\boldsymbol{\tau}(t) \end{aligned} \tag{1}$$

where $\boldsymbol{q}(t)\in\mathbb{R}^{n_d}$ are independent generalized coordinates equal in number to the number of degrees of freedom n_d; $\boldsymbol{v}(t)\in\mathbb{R}^{n_d}$ are the generalized velocities and—using Newton's *dot* notation—$\dot{\boldsymbol{v}}(t)$ contains their time derivatives; $\boldsymbol{\theta}(t)\in\mathbb{R}^{n_p}$ includes system parameters of interest; $M(\boldsymbol{q}(t),\boldsymbol{\theta}(t))\in\mathbb{R}^{n_d\times n_d}$ is the square inertia matrix; $\boldsymbol{C}(\boldsymbol{q}(t),\boldsymbol{v}(t),\boldsymbol{\theta}(t))\in\mathbb{R}^{n_d\times n_d}$ includes centrifugal, gyroscopic, and Coriolis effects; $\boldsymbol{N}(\boldsymbol{q}(t),\boldsymbol{v}(t),\boldsymbol{\theta}(t))\in\mathbb{R}^{n_d}$ are the generalized gravitational and joint forces; $\boldsymbol{S}\in\mathbb{R}^{n_i\times n_d}$ is a selection matrix mapping the applied inputs $\boldsymbol{\tau}(t)\in\mathbb{R}^{n_i}$, or wrenches, appropriately. (For notational brevity, all future equations will drop the explicit time dependence.)

The relationship between the time derivatives of the independent generalized coordinates and the generalized velocities is

$$\dot{\boldsymbol{q}}=\boldsymbol{H}(\boldsymbol{q},\boldsymbol{\theta})\boldsymbol{v} \tag{2}$$

where $\boldsymbol{H}(\boldsymbol{q},\boldsymbol{\theta})$ is a skew-symmetric matrix that is a function of the selected kinematic representation (e.g., Euler angles, Tait–Bryan angles, axis angle, Euler parameters, etc.) [33,80,81]. However, if Eq. (1) is formulated with independent generalized coordinates and the system has a fixed base, as in Refs. [31,32], then Eq. (2) becomes $\dot{\boldsymbol{q}}=\boldsymbol{v}$.

The trajectory of the system is determined by solving Eqs. (1)–(2) as an initial value problem, where $\boldsymbol{q}(0)=\boldsymbol{q}_0$ and $\boldsymbol{v}(0)=\boldsymbol{v}_0$. Also, the system measured outputs are defined by

$$\boldsymbol{y}=\mathcal{O}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\theta}) \tag{3}$$

where $\boldsymbol{y}\in\mathbb{R}^{n_o}$ with n_o equal to the number of outputs.

3 Motion Planning of Deterministic Fully and Underactuated Systems

The task of dynamic system motion planning is a well-studied topic. It aims to determine either a state or input trajectory—or an appropriate combination—to realize some prescribed motion objective. Treatment of fully and underactuated systems presents multiple methodologies for formulating the governing dynamics. The forward dynamics formulation, applicable to both fully and underactuated systems, prescribes actuator inputs that yield state trajectories through numerical integration. The inverse dynamics formulation is the dual to the forward dynamics formulation and is only applicable to fully actuated systems; it has prescribed state trajectories that yield actuator inputs. The inverse dynamics formulation is more computationally efficient as it is only an algebraic evaluation and completely avoids any numerical integration. Finally, the *hybrid dynamics* formulation is applicable to underactuated systems and leverages the benefits of inverse dynamics for actuated joints and relies on forward dynamics for unactuated joints. It prescribes actuated state and unactuated input trajectories to determine unactuated states through numerical integration and actuated inputs through algebraic evaluations. Partitioning the system states and inputs between actuated and unactuated joints in the following manner, $\boldsymbol{q}=\{{}^a\boldsymbol{q},{}^u\boldsymbol{q}\}$ and $\boldsymbol{\tau}=\{{}^a\boldsymbol{\tau},{}^u\boldsymbol{\tau}\}$, facilitates the illustration of what quantities are known versus unknown when using these formulations of the system's dynamics (see Table 1).

Regardless of which dynamics formulation is selected, a common motion planning practice is to approximate infinite dimensional *known* trajectories by a finite-dimensional parameterization [10]. This paper parameterizes all *known* trajectories with B-splines. For example, the parameterization of $\boldsymbol{q}$ takes the form

$$\boldsymbol{q}(\boldsymbol{P},u)=\sum_{i=0}^{n_{sp}}\beta^{i,\wp-1}(u)\boldsymbol{p}^i \tag{4}$$

Table 1 Knowns versus unknowns dynamic properties

Formulation	Known	Unknown
Forward	$\boldsymbol{\tau}$	$\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{v},\dot{\boldsymbol{v}}$
Inverse	$\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{v},\dot{\boldsymbol{v}}$	$\boldsymbol{\tau}$
Hybrid	${}^a\boldsymbol{q},{}^a\dot{\boldsymbol{q}},{}^a\boldsymbol{q},{}^a\boldsymbol{v},{}^a\dot{\boldsymbol{v}},{}^u\boldsymbol{\tau}$	${}^u\boldsymbol{q},{}^u\dot{\boldsymbol{q}},{}^u\boldsymbol{q},{}^u\boldsymbol{v},{}^u\dot{\boldsymbol{v}},{}^a\boldsymbol{\tau}$

and a similar expansion is given for $\tau(\boldsymbol{P},u)$. There are $\left(n_{sp}+1\right)$ control points $\boldsymbol{P}=\left\{\boldsymbol{p}^0,\ldots,\boldsymbol{p}^{n_{sp}}\right\}\in\mathbb{R}^{n_{sp}+1}\times\mathbb{R}^{n_{\dim}}$ with $\boldsymbol{p}^i\in\mathbb{R}^{n_{\dim}}$, where $\boldsymbol{p}^{i,j}$ is the jth element of the ith control point; $m+1$ is nondecreasing knots $u^0\le\cdots\le u^m\in\mathbb{R}$; $\left(n_{sp}+1\right)$ basis $\beta^{i,\wp}(u)$ of degree of $\wp$; and the relation $m=n_{sp}+\wp+1$ must be maintained.

Basis functions $\beta^{i,\wp}(u)$ can be created recursively by the *Cox–de Boor recursion formula:*

$$\beta^{i,0}(u)=\begin{cases}1 & \text{if } u^i\le u<u^{i+1}\\ 0 & \text{otherwise}\end{cases}$$
$$\beta^{i,\wp}(u)=\frac{u-u^i}{u^{i+\wp}-u^i}\beta^{i,\wp-1}(u)+\frac{u^{i+\wp+1}-u}{u^{i+\wp+1}-u^{i+1}}\beta^{i+1,\wp-1}(u) \tag{5}$$

Also, a *clamped* B-spline has $(\wp+1)$ repeated knots at the extremes of the knot list. The clamping allows one to force the curve to be tangent to the first and last control point legs (or segments) at the first and last control points. Meaning, $\tau(\boldsymbol{P},u^0)=\boldsymbol{p}^0$ and $\tau(\boldsymbol{P},u^m)=\boldsymbol{p}^{n_{sp}}$. This enables one to specify the initial and terminal conditions for the curve by the initial and final control points. The remaining interior control points specify the shape of the curve.

Derivatives of B-spline functions are also B-splines. Let $\mathbf{q}'(\boldsymbol{P},u)=(\partial\mathbf{q}(u)/\partial u)$ represent the first derivative of $\mathbf{q}(\boldsymbol{P},u)$. With a slight abuse of Lagrange's derivative notation, let the control points for $\mathbf{q}'(\boldsymbol{P},u)$ be defined as $\boldsymbol{P}'=\left\{\boldsymbol{p}'^0,\ldots,\boldsymbol{p}'^{n_{sp}-1}\right\}$. Unlike $\boldsymbol{P}$, the values of $\boldsymbol{P}'$ are predetermined through the following recursive relation:

$$\boldsymbol{p}'^i=\frac{\wp}{u^{i+\wp+1}-u^{i+1}}\left(\boldsymbol{p}^{i+1}-\boldsymbol{p}^i\right) \tag{6}$$

which gives the $n_{sp}-1$ inherited control points, or $\boldsymbol{P}'\in\mathbb{R}^{n_{sp}-1}\times\mathbb{R}^{n_{\dim}}$. The corresponding $n_{sp}-1$ basis functions $\beta^{i,\wp-1}(u)$ are of degree $\wp-1$ and are also calculated using Eq. (5).

Additionally, all derivative B-splines inherit their knot vector from their *parent* B-spline. However, only a subset of the original knot vector is used. Meaning, the knot vector for a derivative $\boldsymbol{u}'$ is updated by removing the first and last knot from the original knot vector $\boldsymbol{u}$,

$$\boldsymbol{u}'=\{u^1\le\cdots\le u^{m-1}\}\subset\boldsymbol{u} \tag{7}$$

These recursive relations for control points, basis, and knot vectors also apply for higher-order derivatives. Therefore, by defining $\boldsymbol{P}$ for $\boldsymbol{q}(\boldsymbol{P},u)$, all of its derivatives supported by the original degree $\wp$, control points, and knots, are automatically defined [82].

To illustrate, given $\mathbf{q}(\mathbf{P},u)$ defined in Eq. (4), the first and second derivative curves are defined by

$$\mathbf{q}'(\boldsymbol{P}',u')=\sum_{i=0}^{n_{sp}-1}\beta^{i,\wp-1}(u')\boldsymbol{p}'^i \tag{8}$$

$$\mathbf{q}''(\boldsymbol{P}'',u'')=\sum_{i=0}^{n_{sp}-2}\beta^{i,\wp-2}(u'')\boldsymbol{p}''^i \tag{9}$$

Therefore, in order to specify the initial and/or terminal conditions of a derivative clamped B-spline, the slope of the first/last leg of its parent's control points must match the value for the initial/final condition for the derivative. These are determined from Eq. (6).

In a motion planning setting, the knot span $[u^0,u^m)$ can be defined to correspond to the time of a motion plan's trajectory, where $u^0=t_0$ and $u^m=t_f$, or $\beta^{i,\wp}(u)=\beta^{i,\wp}(t)$. Therefore, the curves $\boldsymbol{q}(\boldsymbol{P},u)=\boldsymbol{q}(\boldsymbol{P},t)$ and $\tau(\boldsymbol{P},u)=\tau(\boldsymbol{P},t)$ are defined from $[t_0,t_f)$.

The generalized velocities and accelerations $\boldsymbol{v}(\boldsymbol{P}',t)$ and $\dot{\boldsymbol{v}}(\boldsymbol{P}'',t)$, respectively, may be determined by differentiating Eq. (2), yielding

$$\ddot{\boldsymbol{q}}(\boldsymbol{P}'',t)=\boldsymbol{H}(\boldsymbol{q}(\boldsymbol{P},t),\boldsymbol{\theta})\dot{\boldsymbol{v}}(\boldsymbol{P}'',t)+\boldsymbol{v}(\boldsymbol{P}',t)\left(\frac{\partial\boldsymbol{H}}{\partial t}+\frac{\partial\boldsymbol{H}}{\partial\boldsymbol{q}}\frac{\partial\boldsymbol{q}}{\partial t}+\frac{\partial\boldsymbol{H}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial t}\right) \tag{10}$$

Solving Eq. (2) for $\boldsymbol{v}(\boldsymbol{P}',t)$ and Eq. (10) for $\dot{\boldsymbol{v}}(\boldsymbol{P}'',t)$ yields

$$\boldsymbol{v}(\boldsymbol{P}',t)=(\boldsymbol{H}(\boldsymbol{q}(\boldsymbol{P},t),\boldsymbol{\theta}))^{-1}\dot{\boldsymbol{q}}(\boldsymbol{P}',t) \tag{11}$$

$$\dot{\boldsymbol{v}}(\boldsymbol{P}'',t)=(\boldsymbol{H}(\boldsymbol{q}(\boldsymbol{P},t),\boldsymbol{\theta}))^{-1}\times\left(\ddot{\boldsymbol{q}}(\boldsymbol{P}'',t)-\boldsymbol{v}(\boldsymbol{P}',t)\left(\frac{\partial\boldsymbol{H}}{\partial t}+\frac{\partial\boldsymbol{H}}{\partial\boldsymbol{q}}\frac{\partial\boldsymbol{q}}{\partial t}+\frac{\partial\boldsymbol{H}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial t}\right)\right) \tag{12}$$

The parameterizations (4), and (10)–(12) are equally applicable to appropriate actuated and unactuated subsets.

Once all known trajectories are parameterized the EOMs take on the form

$$Forward:\ \mathcal{F}(\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{v}(\boldsymbol{P}'),\dot{\boldsymbol{v}}(\boldsymbol{P}''),\boldsymbol{\theta})=\boldsymbol{\tau} \tag{13}$$

$$Inverse:\ \boldsymbol{\tau}=\mathcal{F}(\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{v}(\boldsymbol{P}'),\dot{\boldsymbol{v}}(\boldsymbol{P}''),\boldsymbol{\theta}) \tag{14}$$

$$Hybrid:\ \begin{pmatrix}{}^u\dot{\boldsymbol{v}}\\ {}^a\boldsymbol{\tau}\end{pmatrix}=\mathcal{G}({}^a\boldsymbol{q}(\boldsymbol{P}),{}^a\boldsymbol{v}(\boldsymbol{P}'),{}^a\dot{\boldsymbol{v}}(\boldsymbol{P}''),{}^u\boldsymbol{\tau}(\boldsymbol{P}),\boldsymbol{\theta}) \tag{15}$$

where the time dependence has been dropped again for notational convenience.

In the hybrid dynamics case, it is worth mentioning that the unactuated input wrenches ${}^u\boldsymbol{\tau}$ represent joint constraint forces. Depending on the formulation used to determine the EOMs (e.g., analytic versus recursive methods), then ${}^u\boldsymbol{\tau}$ may be implicitly known once $\{{}^a\boldsymbol{q}(\boldsymbol{P}),{}^a\boldsymbol{v}(\boldsymbol{P}),{}^a\dot{\boldsymbol{v}}(\boldsymbol{P})\}$ are specified. In such a formulation Eq. (15) reduces to

$$\begin{pmatrix}{}^u\dot{\boldsymbol{v}}\\ {}^a\boldsymbol{\tau}\end{pmatrix}=\mathcal{G}({}^a\boldsymbol{q}(\boldsymbol{P}),{}^a\boldsymbol{v}(\boldsymbol{P}),{}^a\dot{\boldsymbol{v}}(\boldsymbol{P}),\boldsymbol{\theta}) \tag{16}$$

Once Eqs. (13)–(16) are determined then the NLP-based deterministic motion planning problems may be formulated as

Forward dynamics NLP formulation:

$$\begin{aligned}
&\min_{x=\{\boldsymbol{P}\}} && \mathrm{J}\\
&\text{s.t.} && \textbf{Forward dynamics}\\
& && \mathcal{F}(\boldsymbol{q},\boldsymbol{v},\dot{\boldsymbol{v}},\boldsymbol{\theta})=\boldsymbol{\tau}(\boldsymbol{P})\\
& && \textbf{Kinematics}\\
& && \dot{\boldsymbol{q}}=\boldsymbol{H}(\boldsymbol{q},\boldsymbol{\theta})\boldsymbol{v}\\
& && \textbf{Outputs}\\
& && \boldsymbol{y}=\mathcal{O}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{\theta})\\
& && \textbf{Constraints}\\
& && \mathcal{C}(\boldsymbol{y},\boldsymbol{\tau},\boldsymbol{\theta})\le 0\\
& && \textbf{Hard ICs and TCs conditions}\\
& && \boldsymbol{q}(0)=\boldsymbol{q}_0\\
& && \dot{\boldsymbol{q}}(0)=\dot{\boldsymbol{q}}_0\\
& && \boldsymbol{q}(t_f)=\boldsymbol{q}_{t_f}\\
& && \dot{\boldsymbol{q}}(t_f)=\dot{\boldsymbol{q}}_{t_f}
\end{aligned} \tag{17}$$

Inverse dynamics NLP formulation:

$$
\begin{aligned}
\min_{x=\{P\}} \quad & \mathrm{J} \\
\text{s.t.} \quad & \textbf{Kinematics} \\
& \boldsymbol{v}(\boldsymbol{P}') = (\boldsymbol{H}(\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1}\dot{\boldsymbol{q}}(\boldsymbol{P}') \\
& \dot{\boldsymbol{v}}(\boldsymbol{P}'') = (\boldsymbol{H}(\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1} \\
& \quad \times\left(\ddot{\boldsymbol{q}}(\boldsymbol{P}'') - \boldsymbol{v}(\boldsymbol{P}')\left(\frac{\partial \boldsymbol{H}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial t}\right)\right) \\
& \textbf{Inverse dynamics} \\
& \boldsymbol{\tau} = \mathcal{F}(\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{v}(\boldsymbol{P}'),\dot{\boldsymbol{v}}(\boldsymbol{P}''),\boldsymbol{\theta}) \\
& \textbf{Outputs} \\
& \boldsymbol{y} = \mathcal{O}(\boldsymbol{q}(\boldsymbol{P}),\dot{\boldsymbol{q}}(\boldsymbol{P}'),\boldsymbol{\theta}) \\
& \textbf{Constraints} \\
& \mathcal{C}(\boldsymbol{y},\boldsymbol{\tau},\boldsymbol{\theta}) \le 0 \\
& \textbf{Hard ICs and TCs conditions} \\
& \boldsymbol{q}(0) = \boldsymbol{P}^0 = \boldsymbol{q}_0 \\
& \dot{\boldsymbol{q}}(0) = \boldsymbol{P}'^0 = \dot{\boldsymbol{q}}_0 \\
& \boldsymbol{q}(t_f) = \boldsymbol{P}^{n_{sp}} = \boldsymbol{q}_{t_f} \\
& \dot{\boldsymbol{q}}(t_f) = \boldsymbol{P}'^{n_{sp}-1} = \dot{\boldsymbol{q}}_{t_f}
\end{aligned}
\tag{18}
$$

Hybrid dynamics NLP formulation :

$$
\begin{aligned}
\min_{x=\{P\}} \quad & \mathrm{J} \\
\text{s.t.} \quad & \textbf{Actuated kinematics} \\
& {}^a\boldsymbol{v}(\boldsymbol{P}') = (\boldsymbol{H}({}^a\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1}\,{}^a\dot{\boldsymbol{q}}(\boldsymbol{P}') \\
& {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}'') = (\boldsymbol{H}({}^a\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1} \\
& \quad \times\left({}^a\ddot{\boldsymbol{q}}(\boldsymbol{P}'') - {}^a\boldsymbol{v}(\boldsymbol{P}')\left(\frac{\partial \boldsymbol{H}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial t}\right)\right) \\
& \textbf{Hybrid dynamics} \\
& \begin{pmatrix} {}^u\dot{\boldsymbol{v}} \\ {}^a\boldsymbol{\tau} \end{pmatrix} = \mathcal{G}({}^a\boldsymbol{q}(\boldsymbol{P}),{}^a\boldsymbol{v}(\boldsymbol{P}'),{}^a\dot{\boldsymbol{v}}(\boldsymbol{P}''),{}^u\boldsymbol{\tau}(\boldsymbol{P}),\boldsymbol{\theta}) \\
& \textbf{Unactuated kinematics} \\
& {}^u\dot{\boldsymbol{q}} = {}^u\boldsymbol{H}({}^u\boldsymbol{q},\boldsymbol{\theta})\,{}^u\boldsymbol{v} \\
& \textbf{Outputs} \\
& \boldsymbol{y} = \mathcal{O}(\boldsymbol{q}(\boldsymbol{P}),\dot{\boldsymbol{q}}(\boldsymbol{P}'),\boldsymbol{\theta}) \\
& \textbf{Constraints} \\
& \mathcal{C}(\boldsymbol{y},\boldsymbol{\tau},\boldsymbol{\theta}) \le 0 \\
& \textbf{Hard actuated ICs and TCs conditions} \\
& {}^a\boldsymbol{q}(0) = {}^a\boldsymbol{P}^0 = {}^a\boldsymbol{q}_0 \\
& {}^a\dot{\boldsymbol{q}}(0) = {}^a\boldsymbol{P}^0 = {}^a\dot{\boldsymbol{q}}_0 \\
& {}^a\boldsymbol{q}(t_f) = {}^a\boldsymbol{P}^{n_{sp}} = {}^a\boldsymbol{q}_{t_f} \\
& {}^a\dot{\boldsymbol{q}}(t_f) = {}^a\boldsymbol{P}^{n_{sp}-1} = {}^a\dot{\boldsymbol{q}}_{t_f} \\
& \textbf{Hard unactuated ICs and TCs conditions} \\
& {}^u\boldsymbol{q}(0) = {}^u\boldsymbol{q}_0 \\
& {}^u\dot{\boldsymbol{q}}(0) = {}^u\dot{\boldsymbol{q}}_0 \\
& {}^u\boldsymbol{q}(t_f) = {}^u\boldsymbol{q}_{t_f} \\
& {}^u\dot{\boldsymbol{q}}(t_f) = {}^u\dot{\boldsymbol{q}}_{t_f}
\end{aligned}
\tag{19}
$$

Equations (17)–(19) seek to find the control points $\boldsymbol{P}$ that minimize some prescribed objective function J, while being subject to the dynamic constraints defined in one of Eqs. (13)–(16). Additional constraints may also be defined; for example, maximum/minimum actuator and system parameter limits or physical system geometric limits can be represented as inequality relations $\mathcal{C}(\boldsymbol{y},\boldsymbol{\tau},\boldsymbol{\theta}) \le \boldsymbol{0}$. In the *hybrid dynamics* NLP formulation, Eq. (19) explicitly differentiates between the initial conditions (ICs) and terminal conditions (TCs) for the actuated and unactuated states. All actuated ICs and TCs are determined by corresponding control points in $\boldsymbol{P}$ and all unactuated ICs and TCs are freely defined. When ICs and/or TCs are explicitly defined as shown in Eq. (19) they are referred to as *hard* constraints; conversely, if the constraints are added to the definition of the objective function J, then they are referred to as *soft* constraints.

The literature contains a variety of objective function definitions for J when used in a motion planning setting. Some commonly defined objective functions are

$$J_{D1} = t_f \tag{20}$$

$$J_{D2} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} \tau_i^2(t)dt \tag{21}$$

$$J_{D3} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} |\tau_i(t)\dot{q}_i(t)|dt \tag{22}$$

$$J_{D4} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} \dot{\tau}_i^2(t)dt \tag{23}$$

where Eq. (20) represents a *time optimal* design, Eq. (21) minimizes the effort, Eq. (22) the power, and Eq. (23) the jerk.

The solutions to Eqs. (17)–(19) produce optimal motion plans under the assumption that all system properties are known (i.e., Eqs. (13)–(16) are completely deterministic). The primary contribution of this work is the presentation of variants of Eqs. (17)–(19) that allow Eqs. (13)–(16) to contain uncertainties of diverse types (e.g., parameters, initial conditions, sensor/actuator noise, or forcing functions). The following section will briefly introduce generalized polynomial chaos (gPC) that is used to model the uncertainties and to quantify the resulting uncertain system states and inputs.

4 Generalized Polynomial Chaos

Generalized polynomial chaos (gPC), first introduced by Wiener [34], is an efficient method for analyzing the effects of uncertainties in second order random processes [35]. This is accomplished by approximating a source of uncertainty θ with an infinite series of weighted orthogonal polynomial bases called polynomial chaoses. Clearly an infinite series is impractical, therefore, a truncated set of $p_o + 1$ terms is used with $p_o \in \mathbb{N}$ representing the *order* of the approximation. Or,

$$\theta(\xi) = \sum_{j=0}^{p_o} \theta^j \psi^j(\xi(\omega)) \tag{24}$$

where $\theta^j \in \mathbb{R}$ represent known stochastic coefficients; $\psi^j \in \mathbb{R}$ represent individual single-dimensional orthogonal basis terms (or modes); $\xi(\omega) \in \mathbb{R}$ is the associated random variable for θ that maps the random event $\omega \in \Omega$, from the sample space Ω to the domain of the orthogonal polynomial basis (e.g., $\xi : \Omega \to [-1, 1]$).

Polynomial chaos basis functions are orthogonal with respect to the ensemble average inner product,

$$\left\langle \psi^i(\xi), \psi^j(\xi) \right\rangle = \int_{-1}^{1} \psi^i(\xi)\psi^j(\xi)w(\xi)d\xi = 0, \quad \text{for } i \ne j \tag{25}$$

where $w(\xi)$ is the weighting function that is equal to the joint probability density function of the random variable ξ. Also,

$\langle \psi^j, \psi^j \rangle = 1, \forall j$ when using *normalized basis* and *standardized basis* result in a nonunity constant value and may be efficiently computed off-line using Eq. (25).

Generalized polynomial chaos can be applied to multibody dynamical systems described by differential equations [39,47]. The presence of uncertainty in the system results in uncertain states and/or inputs. Therefore, the uncertain states/inputs can be approximated in a similar fashion as Eq. (24):

$$\dot{v}_i(\xi;t) = \sum_{j=0}^{n_b} \dot{v}_i^j(t)\psi^j(\xi), \quad i = 1,\ldots,n_s \tag{26}$$

$$\tau_i(\xi;t) = \sum_{j=0}^{n_b} \tau_i^j(t)\psi^j(\xi), \quad i = 1,\ldots,n_i \tag{27}$$

where $\dot{v}_i^j(t) \in \mathbb{R}^{n_b}$ represent the gPC expansion coefficients for the ith state; $\tau_i^j(t) \in \mathbb{R}^{n_b}$ represent the gPC expansion coefficients for the ith input; and $n_b \in \mathbb{N}$ representing the number of basis terms in the approximation. It is instructive to notice how time and randomness are decoupled within a single term after the gPC expansion. Only the expansion coefficients are dependent on time, and only the basis terms are dependent on the n_b random variables ξ. Also, any *unknown* itemized in Table 1 has a corresponding approximation as found in Eqs. (26) and (27).

The stochastic basis may be multidimensional in the event there are multiple sources of uncertainty. The multidimensional basis functions are represented by $\Psi^j \in \mathbb{R}^{n_b}$. Additionally, $\boldsymbol{\xi}$ becomes a vector of random variables $\boldsymbol{\xi} = \{\xi_1,\ldots,\xi_{n_p}\} \in \mathbb{R}^{n_p}$, and maps the sample space Ω to an n_p dimensional cuboid $\boldsymbol{\xi} : \Omega \to [-1,1]^{n_p}$ (as in the example of Jacobi chaoses).

The multidimensional basis is constructed from a product of the single-dimensional basis in the following manner:

$$\Psi^j = \psi_1^{i_1}\psi_2^{i_2}\cdots\psi_{n_p}^{i_{n_p}}, \quad i_k = 0,\ldots,p_o, \quad k = 1,\ldots,n_p \tag{28}$$

where subscripts represent the uncertainty source and superscripts represent the associated basis term (or mode). A complete set of basis may be determined from a full tensor product of the single-dimensional bases. This results in an excessive set of $(p_o+1)^{n_p}$ basis terms. Fortunately, the multidimensional sample space can be spanned with a minimal set of $n_b = (n_p+p_o)!/n_p!p_o!$ basis terms. The minimal basis set can be determined by the products resulting from these index ranges:

$$i_1 = 0,\ldots,p_o$$
$$i_2 = 0,\ldots,(p_o - i_1),\ldots$$
$$i_{n_p} = 0,\ldots,\left(p_o - i_1 - i_2 - \cdots - i_{(n_p-1)}\right)$$

The number of multidimensional terms n_b grows quickly with the number of uncertain parameters n_p and polynomial order p_o. Sandu et al. showed that gPC is most appropriate for modeling systems with a relatively low number of uncertainties [39,47] but can handle large nonlinear uncertainty magnitudes.

Substituting Eqs. (24), (26), and (27) into Eqs. (13)–(15) produces the following uncertain dynamics:

***Uncertain forward dynamics (UFD)*:**

$$\mathcal{F}\left(\sum_{j=0}^{n_b}\boldsymbol{q}^j(t)\Psi^j(\xi), \sum_{j=0}^{n_b}\boldsymbol{v}^j(t)\Psi^j(\xi)\sum_{j=0}^{n_b}\dot{\boldsymbol{v}}^j(t)\Psi^j(\xi), \sum_{j=0}^{p_o}\boldsymbol{\theta}_k^j(t)\psi_k^j(\xi_k)\right) = \tau(\boldsymbol{P}) \tag{29}$$

***Uncertain inverse dynamics (UID)*:**

$$\sum_{j=0}^{n_b}\boldsymbol{\tau}^j(t)\Psi^j(\boldsymbol{\xi}) = \mathcal{F}\left(\boldsymbol{q}(\boldsymbol{P}), \boldsymbol{v}(\boldsymbol{P}'), \dot{\boldsymbol{v}}(\boldsymbol{P}''), \sum_{j=0}^{p_o}\boldsymbol{\theta}_k^j(t)\psi_k^j(\xi_k)\right) \tag{30}$$

***Uncertain hybrid dynamics (UHD)*:**

$$\begin{pmatrix} \sum_{j=0}^{n_b} {}^u\dot{v}_i^j(t)\Psi^j(\xi_k) \\ \sum_{j=0}^{n_b} {}^a\tau_l^j(t)\Psi^j(\xi_k) \end{pmatrix} = \mathcal{G}\left({}^a\boldsymbol{q}(\boldsymbol{P}), {}^a\boldsymbol{v}(\boldsymbol{P}'), {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}''), {}^u\boldsymbol{\tau}(\boldsymbol{P}), \sum_{j=0}^{p_o}\boldsymbol{\theta}_k^j(t)\psi^j(\xi_k)\right) \tag{31}$$

where the unknowns are now the unknown gPC expansion coefficients. (Notice how the uncertain parameters are expanded by their associated single-dimensional basis where the dependent states and/or applied inputs are expanded by the multidimensional basis.)

The Galerkin projection method (GPM) is a commonly used method for solving Eqs. (29)–(31), however, this is a very intrusive technique and requires a custom formulation of the dynamic EOMs. As an alternative, sample-based collocation techniques can be used without the need to modify the base EOMs.

Sandu et al. [39,40] showed that the collocation method solves formulations such as Eqs. (29)–(31) by solving Eqs. (13)–(16) at a set of points ${}_k\boldsymbol{\mu} \in \mathbb{R}^{n_p}, k = 1,\ldots,n_{cp}$, selected from the n_p-dimensional domain of the random variables $\boldsymbol{\xi} \in \mathbb{R}^{n_p}$. Meaning, at any given instance in time, the random variables' domain is sampled and solved n_{cp} times with $\boldsymbol{\xi} = {}_k\boldsymbol{\mu}$ (updating the approximations of all sources of uncertainty for each solve), then the uncertain coefficients can be determined at that given time instance. This can be accomplished by defining intermediate variables such as

$${}_k\dot{V}_i(t;{}_k\boldsymbol{\mu}) = \sum_{j=0}^{n_b}\dot{v}_i^j(t)\Psi^j({}_k\boldsymbol{\mu}) \tag{32}$$

$${}_kT_l(t;{}_k\boldsymbol{\mu}) = \sum_{j=0}^{n_b}\tau_l^j(t)\Psi^j({}_k\boldsymbol{\mu}) \tag{33}$$

where $i = 1,\ldots,n_s, k = 0,\ldots,n_{cp}$, and $l = 1,\ldots,n_i$. Substituting them into Eqs. (29)–(31) yields

***Forward dynamics collocation sampling*:**

$${}_k\dot{Q}_i(t;{}_k\mu) = \mathcal{F}({}_kQ_i(t;{}_k\mu), {}_k\Theta_r(t;{}_k\mu)),$$
$$i = 1,\ldots,n_s, k = 0,\ldots,n_{cp}, r = 1,\ldots,n_p \tag{34}$$

***Inverse dynamics collocation sampling*:**

$${}_kT_i(t;{}_k\boldsymbol{\mu}) = \mathcal{F}(\boldsymbol{q}, \boldsymbol{v}, \dot{\boldsymbol{v}}, {}_k\Theta_r(t;{}_k\boldsymbol{\mu})),$$
$$i = 1,\ldots,n_i, k = 0,\ldots,n_{cp}, r = 1,\ldots,n_p \tag{35}$$

***Hybrid dynamics collocation sampling*:**

$$\begin{pmatrix} {}_k^u\dot{V}_i(t;{}_k\boldsymbol{\mu}) \\ {}_k^aT_l(t;{}_k\boldsymbol{\mu}) \end{pmatrix} = \mathcal{G}({}^a\boldsymbol{q}, {}^a\boldsymbol{v}, {}^a\dot{\boldsymbol{v}}, {}^u\boldsymbol{\tau}, {}_k\Theta_r(t;{}_k\boldsymbol{\mu})),$$
$$i = 1,\ldots,{}^un_s, l = 1,\ldots,{}^an_i, k = 0\ldots n_{cp},$$
$$r = 1,\ldots,n_p, \tag{36}$$

where

$${}_k\Theta_r(t;{}_k\boldsymbol{\mu}) = \Sigma_{j=0}^{p_o}\theta_r^j(t)\psi^j({}_k\boldsymbol{\mu}) \tag{37}$$

Equations (34)–(36) provide a set of n_{cp} independent equations whose solutions determine the uncertain expansion coefficients. This is accomplished by recalling the relationship of the expansion coefficients to the solutions as in Eqs. (32) and (33). In matrix notation (32) and (33) can be expressed for all states:

$$\dot{\boldsymbol{V}}_i = (\dot{\boldsymbol{v}}_i(t))^{\boldsymbol{T}}\boldsymbol{\Psi}(\boldsymbol{\mu}), \quad i = 1,\ldots,{}^un_s \tag{38}$$

$$\boldsymbol{T}_l = (\boldsymbol{\tau}_l(t))^T \boldsymbol{\Psi}(\boldsymbol{\mu}), \quad l = 1, \ldots, {}^a n_i \tag{39}$$

where the matrix

$$A_{k,j} = \Psi^j({}_k\mu), \quad j = 0, \ldots, n_b, k = 0, \ldots, n_{cp} \tag{40}$$

is defined as the *collocation matrix*. It is important to note that $n_b \leq n_{cp}$. The expansion coefficients can now be solved for using Eqs. (38) and (39):

$$\dot{v}_i(t) = \boldsymbol{A}^{\#} \dot{\boldsymbol{V}}_i, \quad i = 1, \ldots, {}^u n_s \tag{41}$$

$$\tau_l(t) = \boldsymbol{A}^{\#} \boldsymbol{T}_l, \quad l = 1, \ldots, {}^a n_i \tag{42}$$

where $A^{\#}$ is the pseudo inverse of A if $n_b < n_{cp}$. If $n_b = n_{cp}$, then Eqs. (41) an (42) are simply a linear solve. References [40,48–51] presented the least-squares collocation method (LSCM) where the stochastic state coefficients are solved for, in a least-squares sense, using Eqs. (41) and (42) when $n_b < n_{cp}$. Reference [40] also showed that as $n_{cp} \to \infty$ the LSCM approaches the GPM solution where by selecting $3n_b \leq n_{cp} \leq 4n_b$ the greatest convergence benefit is achieved with minimal computational cost. LSCM also enjoys the same exponential convergence rate as $p_o \to \infty$.

The nonintrusive nature of the LSCM sampling approach is arguably its greatest benefit; Eqs. (13)–(16) may be repeatedly solved without modification. Also, there are a number of methods for selecting the collocation points and the interested reader is recommended to consult Refs. [36–40] for more information.

5 Motion Planning of Uncertain Dynamical Systems

The deterministic motion planning formulations itemized in Eqs. (17)–(19) do not have the ability to account for uncertainties that are inevitably present in a system. The primary contribution of this paper is the development of a new NLP-based framework that, unlike Eqs. (17)–(19) in Sec. 3, directly treats system uncertainties during the motion planning process. Treatment of uncertainties during the motion planning phase allows designers to determine answers to new questions that previously were not possible (or very difficult) to answer.

The following sections present the new uncertain motion planning framework for the forward, inverse, and hybrid dynamics formulations with associated case studies to illustrate the benefits of the framework. It is important to note that while these case studies are simple systems selected intentionally to illustrate the benefits of the new framework, the reader should understand that the framework is general and can be applied to a diversity of problems with uncertainties originating from a variety of sources (e.g. ICs, sensor noise, actuator noise, process noise, parameter uncertainty).

5.1 Forward Dynamics Based Uncertain Motion Planning.

The uncertain motion planning formulation based on forward dynamics is

$$\begin{aligned}
\min_{x=\{\boldsymbol{P}\}} \quad & \mathrm{J} \\
\text{s.t.} \quad & \textbf{Forward dynamics} \\
& \mathcal{F}(\boldsymbol{q}(\xi), \boldsymbol{v}(\xi), \dot{\boldsymbol{v}}(\xi), \boldsymbol{\theta}(\xi)) = \boldsymbol{\tau}(\boldsymbol{P}) \\
& \textbf{Kinematics} \\
& \dot{\boldsymbol{q}}(\xi) = \boldsymbol{H}(\boldsymbol{q}(\xi), \boldsymbol{\theta}(\xi))\boldsymbol{v}(\xi) \\
& \textbf{Outputs} \\
& \boldsymbol{y}(\xi) = \mathcal{O}(\boldsymbol{q}(\xi), \dot{\boldsymbol{q}}(\xi), \boldsymbol{\theta}(\xi)) \\
& \textbf{Constraints} \\
& \mathcal{C}(\boldsymbol{y}(\xi), \boldsymbol{\theta}(\xi), \boldsymbol{\tau}(\boldsymbol{P})) \leq 0 \\
& \textbf{Hard ICs and TCs conditions} \\
& \boldsymbol{q}(0; \xi) = \boldsymbol{q}_0 \\
& \dot{\boldsymbol{q}}(0; \xi) = \dot{\boldsymbol{q}}_0 \\
& \boldsymbol{q}(t_f; \xi) = \boldsymbol{q}_{t_f} \\
& \dot{\boldsymbol{q}}(t_f; \xi) = \dot{\boldsymbol{q}}_{t_f}
\end{aligned} \tag{43}$$

where Eq. (43) is a reformulation of Eq. (17) using the uncertain dynamics defined in Eq. (34).

As illustrated in Table 2, the *known* applied inputs $\boldsymbol{\tau}(\boldsymbol{P})$ are deterministic but the system states $\{\boldsymbol{q}(\xi), \dot{\boldsymbol{q}}(\xi), \ddot{\boldsymbol{q}}(\xi), \boldsymbol{v}(\xi), \dot{\boldsymbol{v}}(\xi)\}$ are uncertain and are modeled using the gPC techniques reviewed in Sec. 4.

Given the deterministic applied inputs in Eq. (43), motion planning may use the deterministic cost functions as defined in Eqs. (20)–(23). However, designs may necessitate statistically penalizing terminal conditions (TC) of the state or output trajectories in the objective function (occasionally referred to as soft constraints). Two candidates are

$$J_{S1} = \left\| \mu_{e(t_f)} \right\| = \left\| E\left[\boldsymbol{e}\left(t_f; \boldsymbol{\xi}\right)\right] \right\| = \left\| \boldsymbol{y}_{\mathrm{ref}}\left(t_f\right) - \boldsymbol{y}^0(t_f) \langle \Psi^0, \Psi^0 \rangle \right\| \tag{44}$$

$$\begin{aligned}
J_{S2} = \left\| \sigma^2_{e(t_f)} \right\| &= \left\| E\left[\left(\boldsymbol{e}\left(t_f; \boldsymbol{\xi}\right) - \mu_{e(t_f)} \right)^2 \right] \right\| \\
&= \left\| \sum_{j=0}^{n_b} \left(\boldsymbol{y}^j(t_f)\right)^2 \langle \Psi^j, \Psi^j \rangle \right\|
\end{aligned} \tag{45}$$

where $\boldsymbol{e}\left(t_f; \boldsymbol{\xi}\right) = \boldsymbol{y}_{\mathrm{ref}}\left(t_f\right) - \boldsymbol{y}(t_f; \boldsymbol{\xi})$. Equation (44) is the expected value of the TC's error and Eq. (45) is the corresponding variance of the TC's error. (Due to the orthogonality of the polynomial basis, Eqs. (44) and (45) result in a reduced set of efficient arithmetic operations on their respective gPC expansion coefficients.) Therefore, when applying soft constraints, the final cost function J may be composed of both deterministic and uncertain terms. For example, $J = a \cdot J_{D3} + b \cdot J_{S1}$, where a and b are scalarization constants.

The inequality constraints may also benefit from added statistical information. For example, bounding the expected values can be expressed as

$$\mathcal{C}(t; \boldsymbol{\xi}) = \underline{\boldsymbol{y}} \leq E[\boldsymbol{y}(\boldsymbol{\xi})] \leq \bar{\boldsymbol{y}} \tag{46}$$

where $E[\boldsymbol{y}(\boldsymbol{\xi})] = \mu_{\boldsymbol{y}} = \boldsymbol{y}^0 \langle \Psi^j, \Psi^j \rangle$, and $\left\{ \underline{\boldsymbol{y}}, \bar{\boldsymbol{y}} \right\}$ are the minimum/maximum output bounds, respectively.

Collision avoidance constraints would ideally involve *supremum* and *infimum* bounds:

$$\underline{\boldsymbol{y}} \leq \inf(\boldsymbol{y}(t; \boldsymbol{\xi})), \quad \sup(\boldsymbol{y}(t; \boldsymbol{\xi})) \leq \bar{\boldsymbol{y}} \tag{47}$$

However, one major difficulty with supremum and infimum bounds is that they are expensive to calculate. A more efficient alternative can be to constrain the uncertain configuration in a weighted standard deviation sense. Collision constraints would then take the form

$$\begin{aligned}
\mu_{y_i} + \alpha_i \cdot \sigma_{y_i} &\leq \bar{y}_i \\
\bar{y}_i &\leq \mu_{y_i} - \alpha_i \cdot \sigma_{y_i}
\end{aligned} \tag{48}$$

where $\operatorname{std}[y_i(\xi)] = \sigma_{y_i} = \sqrt{\sum_{j=1}^{n_b} y_i^j \langle \Psi^j, \Psi^j \rangle}$ and α_i is a linear scaling factor. This gives the practitioner the ability to specify the level of uncertainty to be accounted for in the design in a tunable standard deviation sense.

Table 2 Deterministic knowns versus uncertain unknowns

Formulation	Known ($\boldsymbol{P}$)	Unknown (ξ)
Forward	$\tau(\boldsymbol{P})$	$\boldsymbol{q}(\xi), \dot{\boldsymbol{q}}(\xi), \ddot{\boldsymbol{q}}(\xi), \boldsymbol{v}(\xi), \dot{\boldsymbol{v}}(\xi)$
Inverse	$\boldsymbol{q}(\boldsymbol{P}), \dot{\boldsymbol{q}}(\boldsymbol{P}'), \ddot{\boldsymbol{q}}(\boldsymbol{P}''), \boldsymbol{v}(\boldsymbol{P}'), \dot{\boldsymbol{v}}(\boldsymbol{P}'')$	$\tau(\xi)$
Hybrid	${}^a\boldsymbol{q}(\boldsymbol{P}), {}^a\dot{\boldsymbol{q}}(\boldsymbol{P}'), {}^a\ddot{\boldsymbol{q}}(\boldsymbol{P}''), {}^a\boldsymbol{v}(\boldsymbol{P}'), {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}''), {}^u\tau(\boldsymbol{P})$	${}^u\boldsymbol{q}(\xi), {}^u\dot{\boldsymbol{q}}(\xi), {}^u\ddot{\boldsymbol{q}}(\xi), {}^u\boldsymbol{v}(\xi), {}^u\dot{\boldsymbol{v}}(\xi), {}^a\tau(\xi)$

Case Study. Application of Eq. (43) enables a designer to treat all possible realizations of a given uncertain system when planning motions of fully actuated and underactuated systems. To illustrate, a simple fully actuated serial manipulator "pick-and-place" application (shown in Fig. 1) will now be presented. It is important to reiterate that while the following example illustrates one type of uncertainty, Eq. (43) may be applied to diverse applications with uncertainties unique to that application.

The design objective of this case study is to minimize the effort it takes to move the manipulator from its initial configuration $\boldsymbol{q}_0$ to the target configuration $\boldsymbol{q}_{t_f}$ in a prescribed amount of time t_f. This results in a deterministic objective function of $J = \Sigma_{i=1}^{n_i} z_i \tau^2$, which is frequently referred to as an *effort optimal* design. However, the payload mass $M(\xi)$ is defined to be uncertain rendering the system dynamics uncertain. Since the uncertain serial manipulator is a fully actuated system, where the joints $\boldsymbol{q} = \{q_1, q_2\}$ are actuated with the input wrenches $\boldsymbol{\tau} = \{\tau_1, \tau_2\}$, the motion planning problem may be appropriately defined by Eq. (43).

By parameterizing the input wrench profiles with B-splines, in a similar fashion as Eq. (4), Eq. (43) results in a finite search problem seeking for spline control points $\mathbf{P}$ that minimize the actuation effort defined in J. Therefore, the problem's optimization variables are $\boldsymbol{x} = \{\mathbf{P}\}$.

The actuators are bounded in their torque supply and the manipulator should neither hit the wall it is mounted to nor the obstacle. The constraints may therefore be defined as

$$\mathcal{C}: \begin{cases} \underline{\tau} \leq \boldsymbol{\tau} \leq \bar{\tau} \\ \mu_{y_i} \pm \alpha_i \cdot \sigma_{y_i} \leq \mathbf{0} \\ -\mathfrak{D}_{i,j}(\mu_{\boldsymbol{y}} \pm \alpha^T \sigma_{\boldsymbol{y}}) \leq \mathbf{0} \end{cases} \tag{49}$$

where $i = 1, 2$ and $j =$ obstacle for the signed distance $\mathfrak{D}_{i,j}(\mu_{\boldsymbol{y}} \pm \alpha^T \sigma_{\boldsymbol{y}})$ measured from each link of the serial manipulator to the obstacle calculated using the statistical mean and weighted standard deviations of the configuration/outputs, and $\{\underline{\tau}, \bar{\tau}\}$ are the minimum/maximum input bounds, respectively.

This formulation allows a design engineer to answer the question:

Given actuator and obstacle constraints, what is the "effort optimal" motion plan that accounts for all possible systems within the probability space?

Without accounting for the uncertainty directly in the dynamics and motion planning formulations, design engineers would have a difficult time answering this question.

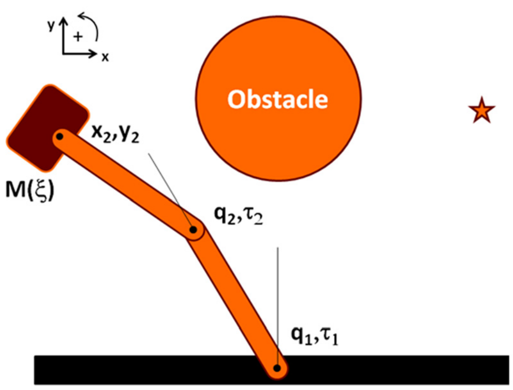

Fig. 1 A simple illustration of an uncertain fully actuated motion planning problem. The forward dynamics based formulation aims to determine an effort optimal motion plan and the inverse dynamics based formulation aims to determine a time optimal motion plan. Both problems are subject to input wrench and geometric collision constraints. This system is an uncertain system due to the uncertain mass of the payload.

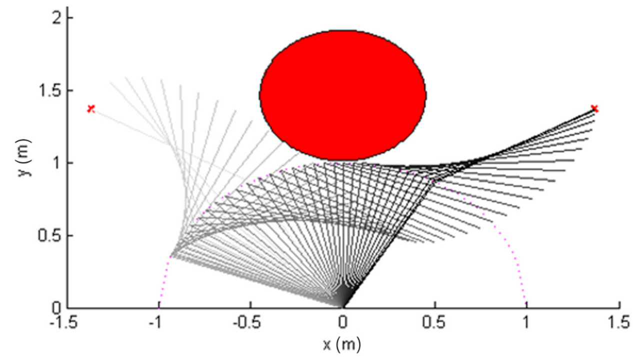

Fig. 2 The effort optimal configuration time histories for the deterministic serial manipulator pick-and-place problem. This optimal solution resulted in a $J = 2770\,(\mathrm{Nm})^2$ design. (The initial configuration starts at the target 'x' on the left and finishes at the 'x' target on the right while progressively darkening from light gray to black.)

The solution to this problem with the deterministic formulation, as defined in Eq. (17), results in an effort optimal solution of $J = 2770\,(\mathrm{Nm})^2$, where $t_f = 1.5$ s. All system parameters are set equal to one, $\theta_i = 1$ (with SI units), initial conditions $\boldsymbol{q}(0) = \{(\pi/6), (\pi/6)\}$ and $\dot{\boldsymbol{q}}(0) = \{0, 0\}$ rad, terminal conditions $\boldsymbol{q}(\mathrm{t_f}) = \{-(\pi/6), -(\pi/6)\}$ and $\dot{\boldsymbol{q}}(\mathrm{t_f}) = \{0, 0\}$ rad, and $\underline{\tau} = -10, \bar{\tau} = 10$ N m. The resulting optimal configuration time history is shown in Fig. 2.

The solution from the new formulation, as defined in Eq. (43) with constraints defined by Eq. (49) with weightings factor $\alpha_i = 1, \forall i = \{1, 2\}$, results in an effort optimal solution of $J = 3530\,(\mathrm{Nm})^2$, where all system parameters and initial/terminal conditions are defined the same as in the deterministic problem. The only difference in this problem definition, as compared to the deterministic problem, is the uncertain payload mass modeled with a uniform distribution having a unity mean and 0.5 variance. The resulting optimal uncertain end-effector Cartesian position time history is illustrated in Fig. 3, where the mean and bounding time histories $\left(\mu_{y_i} \pm \alpha_i \cdot \sigma_{y_i}, \forall i = \{1, 2\}\right)$ are displayed.

Therefore, the effort optimal solution from the uncertain problem resulted in a more conservative answer of $3530\,(\mathrm{Nm})^2$ as compared to $2770\,(\mathrm{Nm})^2$. This is a sensible solution. Close inspection of Fig. 2 shows the deterministic solution drove the configuration as close to the obstacle as possible. The introduction of uncertainty in the payload mass affected the amount of input torque required for the system to reliably avoid the obstacle for all systems within the probability space. In fact, Fig. 3 shows the

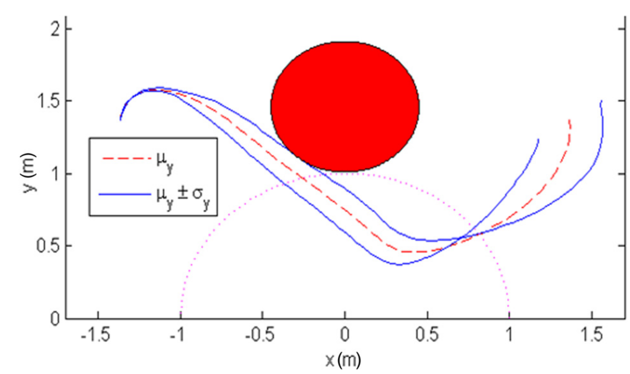

Fig. 3 The effort optimal uncertain end-effector Cartesian position time history for the uncertain serial manipulator pick-and-place problem based on the *uncertain forward dynamics* NLP. The mean and bounding time histories $\mu_{y_i} \pm \alpha_i \cdot \sigma_{y_i}$ are displayed $\forall i = \{1, 2\}$ with $\alpha_i = 1$. This optimal solution resulted in a $J = 3530\,(\mathrm{Nm})^2$ design.

distribution of end-effector Cartesian position trajectory induced by the uncertain payload. The uncertain optimal motion plan from Eq. (43) effectively pushed the end-effector configuration distribution away from the obstacle. This results in a larger effort optimal solution, however, all realizable systems (within a tunable weighted standard deviation sense), are now guaranteed to satisfy the constraints. In other words, the effort optimal solution to Eq. (43) produces the minimum effort design for the entire family of systems (again, within a tunable weighted standard deviation sense). Relying only on the contemporary deterministic problem formulation in Eq. (17) results in an unrealizable trajectory for a subset of the realizable systems.

A third study provides some additional insight to what the new framework can provide. By redefining the objective function for Eqs. (43) as (45) the uncertain design is no longer an effort optimal but *terminal variance optimal* design. In other words, the new design question is:

Given actuator and obstacle constraints, what motion plan will minimize the variance of the terminal condition's (TC) error when accounting for all possible systems within the probability space?

The effort optimal design resulted in a TC error standard deviation of $\sigma_{e(t_f)} = [0.191, 0.133]$ m. Redesigning the motion plan using an objective function defined by Eq. (45) results in a TC error standard deviation of $\sigma_{e(t_f)} = [0.144, 0.114]$ m, as shown in Fig. 4. Therefore, a modest reduction in the TC error standard deviation was realized, however, the effort of the new design increased from 3530 to 5910 $(\mathrm{Nm})^2$. These results indicate a Pareto optimal trade-off between the effort and TC's variance. Therefore, designers may define a hybrid objective function with a scalarization between the effort optimal and terminal variance optimal terms.

One additional insight gained from the terminal variance optimal design is related to the controllability of an uncertain system's TC variance. If the TC variance was fully controllable then the terminal variance optimal design would be able to reduce it to zero. This initial investigation indicates that the variance is not fully controllable. A rigorous uncertain system controllability investigation is out of the scope of this work but will be considered for future research.

A final observation is that the *uncertain forward dynamics* motion planning framework embodied in Eq. (43) is most applicable to force controlled systems where input wrenches are prescribed. However, configuration/position controlled systems may be better designed through application of the uncertain inverse dynamics based NLP found in Eq. (50); this is illustrated in the next section.

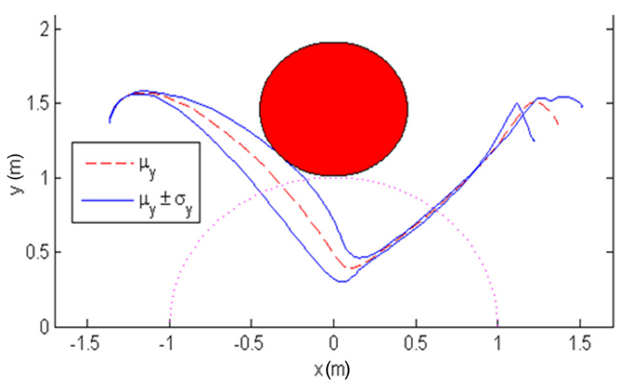

Fig. 4 The terminal variance optimal uncertain end-effector Cartesian position time history for the uncertain serial manipulator pick-and-place problem based on the uncertain forward dynamics NLP. The mean and bounding time histories $\mu_{y_i} \pm \alpha_i \cdot \sigma_{y_i}$ are displayed $\forall i = \{1, 2\}$ with $\alpha_i = 1$. This optimal solution resulted in a $J = 5910\,(\mathrm{Nm})^2$ design.

5.2 Inverse Dynamics Based Uncertain Motion Planning.

The uncertain motion planning formulation based on inverse dynamics is

$$\begin{aligned}
\min_{x=\{P\}} \quad & J \\
\text{s.t.} \quad & \textbf{Kinematics} \\
& v(P') = (H(q(P),\theta))^{-1}\dot{q}(P') \\
& \dot{v}(P'') = (H(q(P),\theta))^{-1} \\
& \quad \times \left(\ddot{q}(P'') - v(P')\left(\frac{\partial H}{\partial t} + \frac{\partial H}{\partial q}\frac{\partial q}{\partial t} + \frac{\partial H}{\partial \theta}\frac{\partial \theta}{\partial t}\right)\right) \\
& \textbf{Inverse dynamics} \\
& \tau(\xi) = \mathcal{F}(q(P), v(P'), \dot{v}(P''), \theta(\xi)) \\
& \textbf{Outputs} \\
& y(\xi) = \mathcal{O}(q(P), \dot{q}(P'), \theta(\xi)) \\
& \textbf{Constraints} \\
& \mathcal{C}(y(\xi), \theta(\xi), \tau(\xi)) \le 0 \\
& \textbf{Hard ICs and TCs conditions} \\
& q(0) = P^0 = q_0 \\
& \dot{q}(0) = P'^0 = \dot{q}_0 \\
& q(t_f) = P^{n_{sp}} = q_{t_f} \\
& \dot{q}(t_f) = P'^{n_{sp}-1} = \dot{q}_{t_f}
\end{aligned} \tag{50}$$

where Eq. (50) is a reformulation of Eq. (18) using the uncertain dynamics defined in Eq. (35).

As illustrated in Table 2, the known state trajectories $\{q(P), \dot{q}(P'), \ddot{q}(P''), v(P'), \dot{v}(P'')\}$ are deterministic but the applied inputs $\tau(\xi)$ are uncertain and are modeled using the gPC techniques reviewed in Sec. 4.

Given the uncertain applied inputs, the most interesting part of Eq. (50) comes in the definition of the objective function terms and constraints. These terms now have the ability to approach the design accounting for uncertainties by way of expected values, variances, and standard deviations. Recalling the definitions of an expected value and variance, Eqs. (21)–(23) may be redefined statistically:

$$\begin{aligned}
J_{S3} &= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f} E\left[\mathbf{z}_i(\tau_i(\boldsymbol{\xi},t))^2\right]dt \\
&= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f}\sum_{j=0}^{n_b}\mathbf{z}_i\left(\tau_i^j(t)\right)^2\langle\Psi^j,\Psi^j\rangle dt
\end{aligned} \tag{51}$$

$$\begin{aligned}
J_{S4} &= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f} E[|\mathbf{z}_i\tau_i(\boldsymbol{\xi},t)y_i(\boldsymbol{\xi},t)|]dt \\
&= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f}\sum_{j=0}^{n_b}\left|\mathbf{z}_i\tau_i^j(t)y_i^j(t)\langle\Psi^{\mathrm{j}},\Psi^{\mathrm{j}}\rangle\right|dt
\end{aligned} \tag{52}$$

$$\begin{aligned}
J_{S5} &= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f} E\left[\mathbf{z}_i(\dot{\tau}_i(\boldsymbol{\xi},t))^2\right]dt \\
&= \sum_{i=1}^{n_i}\int_{t_0=0}^{t_f}\sum_{j=0}^{n_b}\mathbf{z}_i\left(\dot{\tau}_i^j(t)\right)^2\langle\Psi^{\mathrm{j}},\Psi^{\mathrm{j}}\rangle dt
\end{aligned} \tag{53}$$

where $\mathbf{z}$ is a vector of (optional) scalarization weights. The function (51) defines the expected effort, Eq. (52) the expected power with $y_i(\xi,t) = \dot{q}_i(\xi,t)$, and Eq. (53) the expected jerk. Close inspection of Table 2 shows that these statistically based objective function terms are applicable to the inverse and hybrid dynamics based motion planning formulations (50) and (55).

Case Study. Application of Eq. (50) enables a designer to treat all possible realizations of a given uncertain system when planning motions of fully actuated systems. To illustrate we will reuse the simple fully actuated serial manipulator pick-and-place application presented in Sec. 5.1 as shown in Fig. 1.

The design objective for this example is to minimize the time it takes to move the manipulator from its initial configuration $\boldsymbol{q}_0$ to the target configuration $\boldsymbol{q}_{t_f}$. This results in a deterministic objective function $J = t_f$, which is frequently referred to as a *time optimal* design. However, the payload mass $M(\xi)$ is defined to be uncertain rendering the system dynamics uncertain. Since the uncertain serial manipulator is a fully actuated system, where the joints $\boldsymbol{q} = \{q_1, q_2\}$ are actuated with the input wrenches $\boldsymbol{\tau} = \{\tau_1, \tau_2\}$, the motion planning problem may be appropriately defined by (50).

By parameterizing the deterministic joint trajectories with B-splines, as in Eq. (4), Eq. (50) results in a finite search problem seeking for spline control points $\mathbf{P}$ that minimize the trajectory time t_f. Therefore, the problem's optimization variables are $\boldsymbol{x} = \{\mathbf{P}, t_f\}$.

The actuators are bounded in their torque supply and the manipulator should neither hit the wall it is mounted to nor the obstacle. The constraints may therefore be defined as

$$\mathcal{C} : \begin{cases} \mu_{\tau_i} + \alpha_i \cdot \sigma_{\tau_i} \leq \bar{\tau} \\ \underline{\tau} \leq \mu_{\tau_i} - \alpha_i \cdot \sigma_{\tau_i} \\ -\mathrm{y}_1 \leq 0 \\ -\mathrm{y}_2 \leq 0 \\ -\mathfrak{D}_{i,j} \leq 0 \end{cases} \tag{54}$$

where $i = 1, 2$ and $j =$ obstacle for the signed distance $\mathfrak{D}_{i,j}$ measured from each link of the serial manipulator to the obstacle.

Notice the bounding constraints on the input wrenches are defined by their statistical mean and weighted standard deviations, as in Eq. (48), to quantify their uncertainty. Ideally these constraints would be defined by the extremes of the wrench distribution (i.e., the supremum and the infimum), however, due to their computational complexity the approximation by the mean and tunable weighted standard deviation is used.

Since the state trajectories are deterministic, the signed obstacle avoidance constraints $-\mathfrak{D}_{i,j} \leq 0$ and Cartesian wall avoiding constraints $-y_1, -y_2 \leq 0$ are deterministically defined.

This formulation allows a design engineer to answer the question:

Given actuator and obstacle constraints, what is the "time optimal" motion plan that accounts for all possible systems within the probability space?

Without accounting for the uncertainty directly in the dynamics and motion planning formulations, design engineers would have a difficult time answering this question.

The solution to this problem with the deterministic formulation, as defined in Eq. (18), results in a time optimal solution of $t_f = 1.12$ s where all system parameters are set equal to one, $\theta_{\mathrm{i}} = 1$ (with SI units), with initial conditions $\boldsymbol{q}(0) = \{(\pi/6), (\pi/6)\}$ and $\dot{\boldsymbol{q}}(0) = \{0, 0\}$ rad, terminal conditions $\boldsymbol{q}(t_f) = \{-(\pi/6), -(\pi/6)\}$ and $\dot{\boldsymbol{q}}(t_f) = \{0,0\}$ rad, and $\underline{\tau} = -10, \bar{\tau} = 10$ N m. The resulting optimal input wrench time history is shown in Fig. 5.

The solution from the new formulation, as defined in Eq. (50) with constraints defined by Eq. (54) and weighting factor $\alpha_i = 1$, $\forall i = \{1, 2\}$, results in a time optimal solution of $t_f = 1.2$ s, where all system parameters and initial/terminal conditions are defined the same as in the deterministic problem. The only difference in this problem definition when compared to the deterministic problem is the uncertain payload mass is modeled with a uniform distribution having a 1 kg mean and 0.5 kg standard deviation. The resulting optimal uncertain input wrench time history is illustrated in Fig. 6, where each input wrench is displaying its mean value and bounding time histories $\mu_{\tau_i} \pm \alpha_i \cdot \sigma_{\tau_i}, \forall i = \{1, 2\}$. Also, the resulting configuration time history for the optimal uncertain motion plan is shown in Fig. 7.

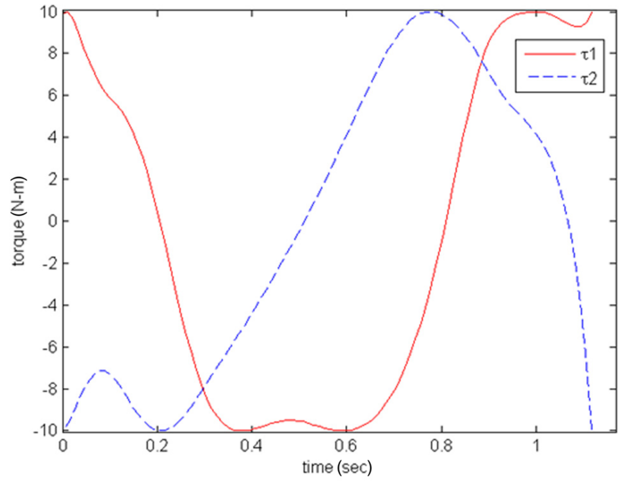

Fig. 5 The time optimal input wrench time histories for the deterministic serial manipulator pick-and-place problem based on the uncertain inverse dynamics NLP. This optimal solution resulted in a $t_f = 1.12$ s.

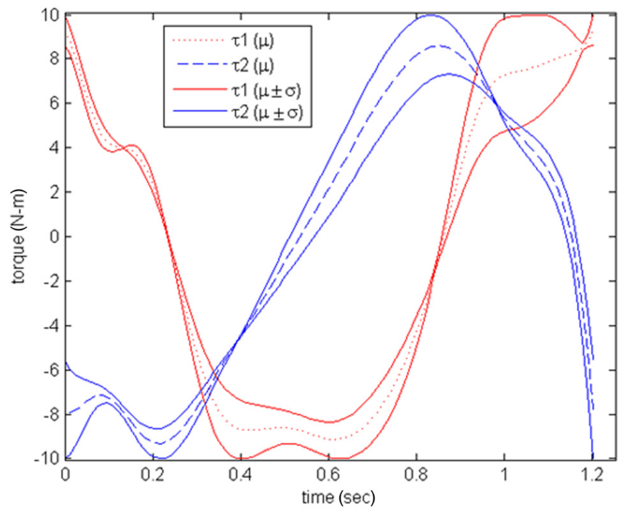

Fig. 6 The time optimal uncertain input wrench time histories for the uncertain serial manipulator pick-and-place problem based on the uncertain inverse dynamics NLP. Each input wrench is displaying its mean value and bounding time histories $\mu_{\tau_i} \pm \alpha_i \cdot \sigma_{\tau_i}$ with $\alpha_i = 1, \forall i = \{1, 2\}$. This optimal solution resulted in a $t_f = 1.2$ s.

Therefore, the time optimal solution from the uncertain problem resulted in a more conservative answer (1.2 s as compared to 1.12 s). This is a sensible solution. Close inspection of Fig. 5 shows the deterministic solution drove the input wrenches to their extreme bounds of ± 10 N m at certain points during the motion profile. Clearly introducing the uncertain mass to the system affected the amount of input torque required for the system to reliably follow the specified state trajectory. In fact, Fig. 6 shows the distribution of input wrenches induced by the uncertain mass. The uncertain optimal motion plan from Eq. (50) effectively pushed the input wrench distribution inside the actuation limits $\{\underline{\tau}, \bar{\tau}\}$. This results in a slower time optimal solution, however, all realizable systems (within a tunable weighted standard deviation sense), are now guaranteed to satisfy the constraints. In other words, the time optimal solution to Eq. (50) produces the minimum time for the entire family of systems (again, within a tunable "weighted standard deviation sense"). Relying only on the contemporary deterministic problem formulation in Eq. (18) results in an unrealizable trajectory for a subset of the realizable systems.

A final observation is that the uncertain inverse dynamics motion planning framework embodied in Eq. (50) is most applicable to configuration/position controlled systems, where states are prescribed as they are in Eq. (4). However, force controlled systems may be better designed through application of Eq. (43) based on uncertain forward dynamics as illustrated in the previous section, Sec. 5.1.

5.3 Hybrid Dynamics Based Uncertain Motion Planning. The uncertain motion planning formulation based on hybrid dynamics is

$$
\begin{aligned}
&\min_{x=\{P\}} \quad J\\
&\text{s.t.} \quad \textbf{Actuated kinematics}\\
&\quad {}^a\boldsymbol{v}(\boldsymbol{P}') = (\boldsymbol{H}({}^a\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1}\,{}^a\dot{\boldsymbol{q}}(\boldsymbol{P}')\\
&\quad {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}'') = (\boldsymbol{H}({}^a\boldsymbol{q}(\boldsymbol{P}),\boldsymbol{\theta}))^{-1}\\
&\qquad \times\left({}^a\ddot{\boldsymbol{q}}(\boldsymbol{P}'') - {}^a\boldsymbol{v}(\boldsymbol{P}')\left(\frac{\partial \boldsymbol{H}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\theta}}\frac{\partial \boldsymbol{\theta}}{\partial t}\right)\right)\\
&\quad \textbf{Hybrid dynamics}\\
&\quad \begin{pmatrix} {}^u\dot{\boldsymbol{v}}(\xi) \\ {}^a\boldsymbol{\tau}(\xi) \end{pmatrix} = \mathcal{G}({}^a\boldsymbol{q}(\boldsymbol{P}), {}^a\boldsymbol{v}(\boldsymbol{P}'), {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}''), {}^u\boldsymbol{\tau}(\boldsymbol{P}), \boldsymbol{\theta}(\xi))\\
&\quad \textbf{Unactuated kinematics}\\
&\quad {}^u\dot{\boldsymbol{q}}(\xi) = {}^u\boldsymbol{H}({}^u\boldsymbol{q}(\xi),\boldsymbol{\theta}(\xi))\,{}^u\boldsymbol{v}(\xi)\\
&\quad \textbf{Outputs}\\
&\quad \boldsymbol{y}(\xi) = \mathcal{O}(\boldsymbol{q}(\boldsymbol{P};\xi), \dot{\boldsymbol{q}}(\boldsymbol{P}';\xi), \boldsymbol{\theta}(\xi))\\
&\quad \textbf{Constraints}\\
&\quad \mathcal{C}(\boldsymbol{y}(\xi),\boldsymbol{\theta}(\xi),\boldsymbol{\tau}(\xi)) \le 0\\
&\quad \textbf{Hard actuated ICs and TCs conditions}\\
&\quad {}^a\boldsymbol{q}(0) = {}^a\boldsymbol{P}^0 = {}^a\boldsymbol{q}_0\\
&\quad {}^a\dot{\boldsymbol{q}}(0) = {}^a\boldsymbol{P}^0 = {}^a\dot{\boldsymbol{q}}_0\\
&\quad {}^a\boldsymbol{q}(t_f) = {}^a\boldsymbol{P}^{n_{sp}} = {}^a\boldsymbol{q}_{t_f}\\
&\quad {}^a\dot{\boldsymbol{q}}(t_f) = {}^a\boldsymbol{P}'^{n_{sp}-1} = {}^a\dot{\boldsymbol{q}}_{t_f}\\
&\quad \textbf{Hard unactuated ICs and TCs conditions}\\
&\quad {}^u\boldsymbol{q}(0;\xi) = {}^u\boldsymbol{q}_0(\xi)\\
&\quad {}^u\dot{\boldsymbol{q}}(0;\xi) = {}^u\dot{\boldsymbol{q}}_0(\xi)\\
&\quad {}^u\boldsymbol{q}(t_f;\xi) = {}^u\boldsymbol{q}_{t_f}(\xi)\\
&\quad {}^u\dot{\boldsymbol{q}}(t_f;\xi) = {}^u\dot{\boldsymbol{q}}_{t_f}(\xi)
\end{aligned}
\tag{55}
$$

where Eq. (55) is a reformulation of Eq. (19) using the uncertain dynamics defined in (36).

As illustrated in Table 2, the actuated states $\{{}^a\boldsymbol{q}(\boldsymbol{P}), {}^a\dot{\boldsymbol{q}}(\boldsymbol{P}'), {}^a\ddot{\boldsymbol{q}}(\boldsymbol{P}''), {}^a\boldsymbol{v}(\boldsymbol{P}'), {}^a\dot{\boldsymbol{v}}(\boldsymbol{P}'')\}$ and the unactuated inputs ${}^u\boldsymbol{\tau}(\boldsymbol{P})$ are known deterministic quantities. Conversely, the unactuated system states $\{{}^u\boldsymbol{q}(\xi), {}^u\dot{\boldsymbol{q}}(\xi), {}^u\ddot{\boldsymbol{q}}(\xi), {}^u\boldsymbol{v}(\xi), {}^u\dot{\boldsymbol{v}}(\xi)\}$ and actuated inputs ${}^a\boldsymbol{\tau}(\xi)$ are *uncertain* and are modeled using the gPC techniques reviewed in Sec. 4.

Case Study. The simple underactuated inverting double pendulum problem (shown in Fig. 8) was selected to illustrate benefits of the uncertain motion planning based on the hybrid dynamics formulation (55). The design objective of this example is to minimize the power it takes to move the manipulator from its initial hanging configuration $\boldsymbol{q}_0$ to the target inverted configuration $\boldsymbol{q}_{t_f}$. The double pendulum is an underactuated system, where only joint q_1 is actuated (by input τ_1), and the mass of the second link is uncertain; therefore, the motion planning problem may be appropriately defined by Eq. (55).

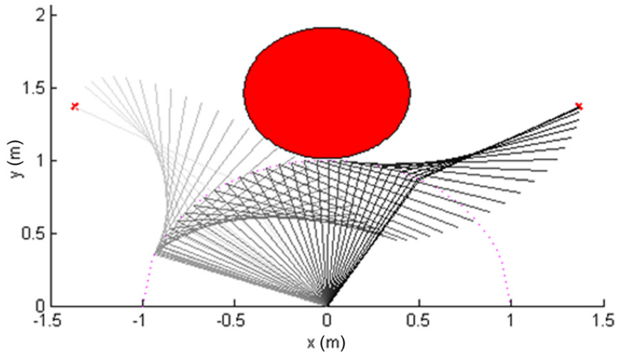

Fig. 7 The final optimal configuration time history of the uncertain serial manipulator pick-and-place application involving collision avoidance and actuator constraints design with the uncertain inverse dynamics NLP. (The initial configuration starts at the target 'x' on the left and finishes at the 'x' target on the right while progressively darkening from light gray to black.)

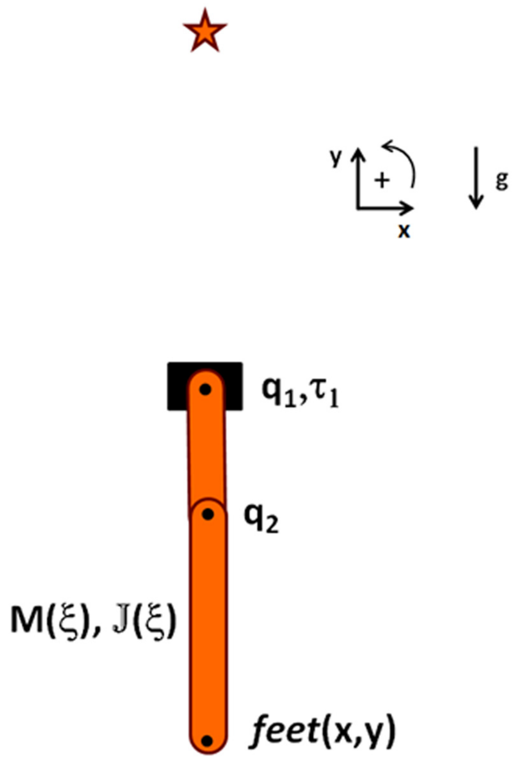

Fig. 8 A simple illustration of the underactuated uncertain hybrid dynamics motion planning formulation. This problem aims to determine a power optimal motion plan to lift the pendulum from the initial hanging configuration to an inverted vertical configuration when subject to input wrench and terminal condition constraints. This is an uncertain system due to the mass of the payload.

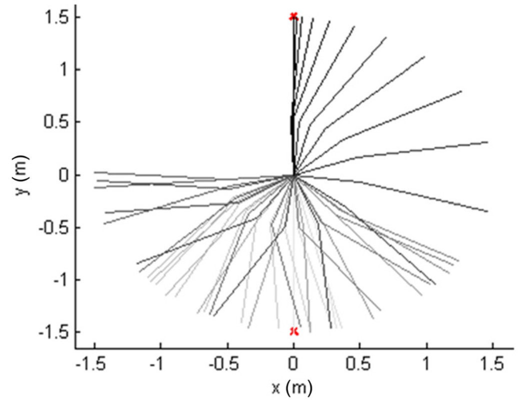

Fig. 9 The power optimal configuration time history for the deterministic inverting double pendulum. This optimal solution resulted in a 1060 W design. (The initial configuration starts with the double pendulum in the down position and swings up to the vertical while progressively darkening from light gray to black.)

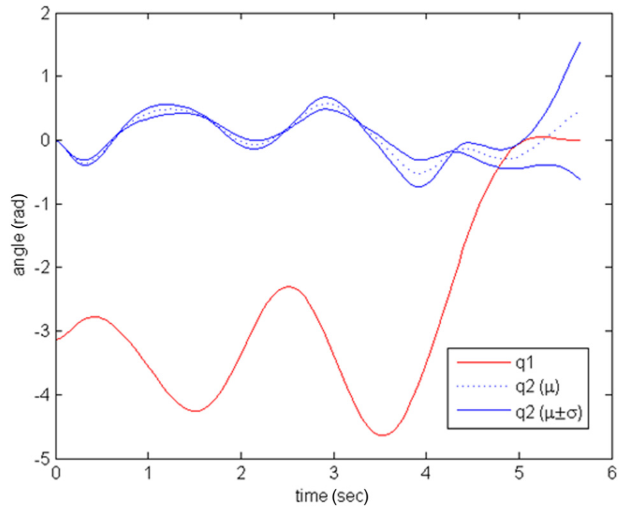

Fig. 11 The joint time histories for the deterministically design motion plan applied to an uncertain inverting double pendulum (where $\mu_{q2} \pm \alpha\sigma_{q2}$ with $\alpha = 1$). The presence of the uncertainty results in the expected terminal error condition not being satisfied with excessive variance.

By parameterizing the actuated state profiles with B-splines, as in Eq. (4), and using the hybrid dynamics defined in Eq. (16), Eq. (55) results in a finite search problem seeking for spline control points $\mathbf{P}$ and terminal time t_f that minimize the system's power. Therefore, the problem's optimization variables are $\boldsymbol{x} = \{\mathbf{P}, t_f\}$. Assuming a soft terminal error expected value condition is used, the objective function becomes $J = a \cdot J_{S1} + b \cdot J_{S4}$, where a and b are scalarization constants.

The actuators are bounded in their torque supply. Additionally, suppose the design has a specified variance in the terminal error conditions (45) that must be satisfied. Implementing both of these design constraints as *hard* constraints takes the form

$$\mathcal{C} : \begin{cases} \underline{\tau} \le \tau \le \bar{\tau} \\ \sigma^2_{e(t_f)} \le \bar{\sigma}^2_{e(t_f)} \end{cases} \tag{56}$$

where $\{\underline{\tau}, \bar{\tau}\}$ are the minimum/maximum input bounds, respectively, and $\bar{\sigma}^2_{e(t_f)}$ is the maximum acceptable terminal error variance.

This formulation allows a design engineer to answer the question:

Given actuator and terminal error variance constraints, what motion plan will minimize the system's power over the trajectory when accounting for all possible systems within the probability space?

Without accounting for the uncertainty directly in the dynamics and motion planning formulations, design engineers would have a difficult time answering this question.

The solution to this problem with the deterministic formulation, as defined in Eq. (19), results in an *power optimal* solution of $J_{S1} = 1060\mathrm{W}$ with $t_f = 5.66$ s, where all system parameters are set equal to $\theta_i = 1$ (with SI units) except the length of the first link is set to 0.5 m, initial conditions $\boldsymbol{q}(0) = \{-\pi, 0\}$ and $\dot{\boldsymbol{q}}(0) = \{0, 0\}$ rad, terminal conditions $\boldsymbol{q}(t_f) = \{0, 0\}$ and $\dot{\boldsymbol{q}}(t_f) = \{0, 0\}$ rad, and the input limits are $\underline{\tau} = -10, \bar{\tau} = 10\,\mathrm{Nm}$. The resulting optimal motion plan's configuration time history is shown in Fig. 9.

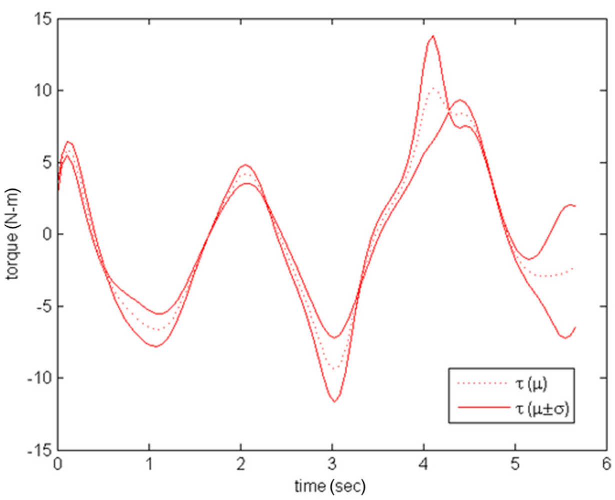

Fig. 10 The uncertain input wrench time history for the deterministically designed motion plan applied to an uncertain inverting double pendulum (where $\mu_\tau \pm \alpha\sigma_\tau$ with $\alpha = 1$). The presence of the uncertainty results in both the maximum and minimum input limits being exceeded.

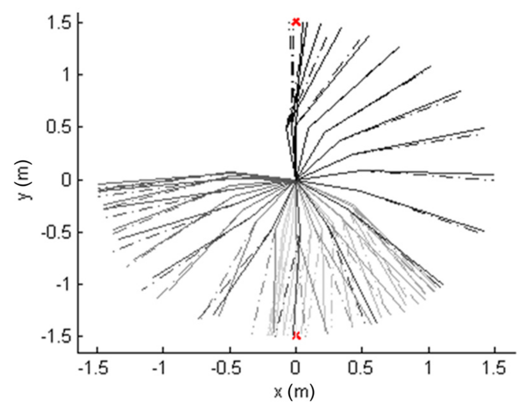

Fig. 12 The power optimal configuration time history for the uncertain inverting double pendulum based on uncertain hybrid dynamics NLP, where $\{\mu_{y_i} - \alpha_i \cdot \sigma_{y_i}(\text{solid}), \mu_{y_i} + \alpha_i \cdot \sigma_{y_i}(\text{dash} - \text{dot})\}, \forall i = \{1, 2\}$ with $\alpha_i = 1$. This optimal solution resulted in a 310 W design. (The initial configuration starts with the double pendulum in the down position and swings up to the vertical while progressively darkening from light gray to black.)

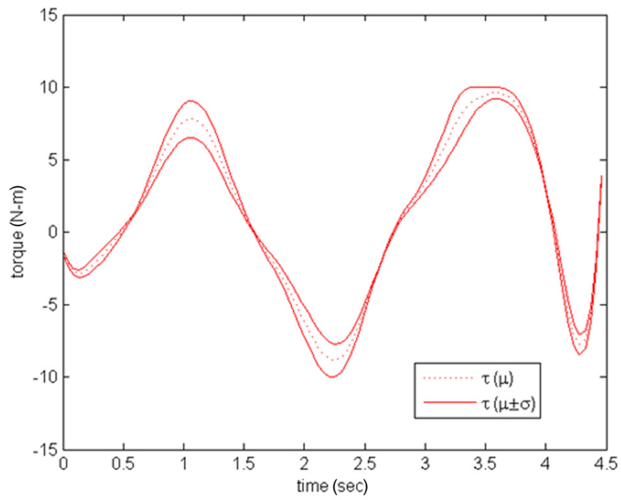

Fig. 13 The uncertain input wrench time history resulting from the motion plan generated by the new uncertain hybrid dynamics NLP (where $\mu_\tau \pm \alpha\sigma_\tau$ with $\alpha = 1$). Both the maximum and minimum input limits were satisfied, in a weighted standard deviation sense, for all systems within the probability space.

The value of the new framework is best illustrated by applying the deterministically designed motion profile to an uncertain system. Figures 10 and 11 show the results of the deterministic motion plan applied to a system with a single uncertainty where the second link has an uncertain mass with $\mu_{m2} = 1\,\text{kg}$ and $\sigma^2_{m2} = 0.5\,\text{kg}^2$. Figure 10 shows that the resulting input profile exceeds both the upper and lower bounding constraints of $\underline{\tau} = -10, \bar{\tau} = 10$ N m. Additionally, Fig. 11 shows that the target terminal configuration was not satisfied and an excessive terminal error variance is experienced.

Approaching the design with the new framework accounts for the uncertainties up front during the optimal search and results in a design that satisfies all constraints for all possible systems in the probability space. This is accomplished by application of Eq. (55) with constraints defined by Eq. (56), where $\bar{\sigma}^2_{e(t_f)} = 0.01\ \text{m}^2$. This results in a power optimal solution of $J_{S2} = 310\,\text{W}$ with $t_f = 4.46$ s (where the same uncertain second link mass is reused). The resulting motion plan's optimal uncertain configuration time history is illustrated in Fig. 12 where the bounding configuration time histories $\{\mu_{y_i} - \alpha_i \cdot \sigma_{y_i}(\text{solid}), \mu_{y_i} + \alpha_i \cdot \sigma_{y_i}(\text{dash} - \text{dot})\}$ are displayed $\forall i = \{1, 2\}$ with $\alpha_i = 1$. The Euclidean norm of the soft expected value terminal configuration constraint was very acceptable, $E[\boldsymbol{e}(t_f)] = 2.61 \times 10^{-6}$ m. Figure 13 shows that the input wrench constraints for the entire probability space were satisfied (within a tunable weighted standard deviation sense). Figure 14 show that the specified terminal error variance was also satisfied, $\sigma^2_{e(t_f)} = 0.00321 \leq \bar{\sigma}^2_{e(t_f)} = 0.01\text{m}^2$.

The reduced power of the uncertain design, as compared to the deterministic design, makes sense in that the expected input values $E[\tau_1]$ of the uncertain design (as shown in Fig. 13), are lower than those in the deterministic design (as shown in Fig. 10). This relationship is also true for $\dot{q}_1$ (although are not illustrated), therefore, the product of the reduced expected torque and joint rate yields a lower system power.

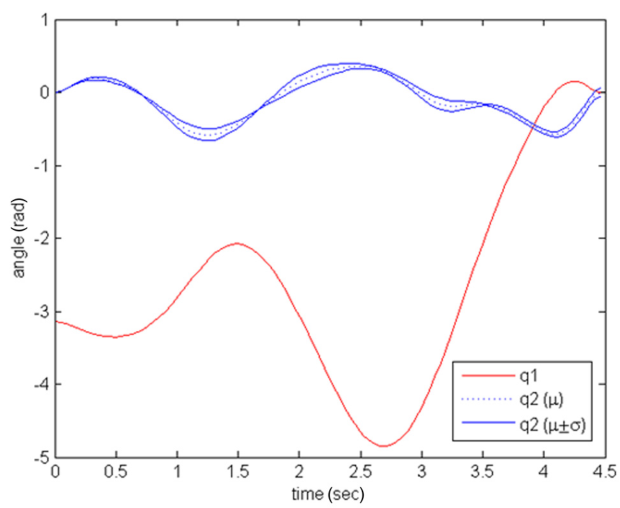

Fig. 14 The joint time histories resulting from the motion plan generated by the new uncertain hybrid dynamics NLP (where $\mu_{q2} \pm \alpha\sigma_{q2}$ with $\alpha = 1$). The resulting terminal error variance satisfies the specification $\sigma^2_{e(t_f)} = 0.0032 \leq \sigma^2_{e(t_f)} = 0.01\text{m}^2$.

6 Conclusions

This work has presented a new nonlinear programming based framework for motion planning that treats uncertain fully actuated and underactuated dynamical systems described by ordinary differential equations. The framework allows practitioners to model sources of uncertainty using the generalized polynomial chaos methodology and to solve the uncertain forward, inverse, and hybrid dynamics using a least-squares collocation method. Subsequently, statistical information from the uncertain dynamics may be included in the NLP's objective function and constraints to perform optimal motion planning under uncertainty. Three case studies with uncertain dynamics illustrate how the new framework produces an optimal design that accounts for the entire family of systems within the associated probability space. This adds robustness to the design of the optimally performing system.

In future work the authors will expand the new framework to treat constrained dynamical systems described by differential algebraic equations.

Acknowledgment

This work was partially supported by the Automotive Research Center (ARC), Thrust Area 1.

Nomenclature

General

$x, X =$ nonbolded variables indicate a scalar quantity
$\boldsymbol{x}, \boldsymbol{X} =$ bolded lower case variables are vectors, upper case variables are matrices
$x_i =$ bottom right index indicates a state
$x^j =$ top right index indicates a stochastic coefficient, or mode
${}_k x =$ bottom left index associates x to a specific collocation point
${}^a x, {}^u x =$ top left annotations indicate if a given variable is *actuated* or *unactuated*
${}^u_k x^j_i =$ the four major variable annotations
$\underline{x}, \bar{x} =$ lower and upper bounds on x
$\xi =$ random variable

Indexes and Dimensions

$m \in \mathbb{N} =$ number of B-spline knots
$n_b \in \mathbb{N} =$ number of multidimensional basis terms
$n_{cp} \in \mathbb{N} =$ number of collocation points
$n_d \in \mathbb{N} =$ number of degrees-of-freedom (DOF)
$n_{\dim} \in \mathbb{N} =$ number of dimensions of the B-spline (e.g., n_d or n_i)
$n_i \in \mathbb{N} =$ number of input wrenches, $\boldsymbol{\tau} \in \mathbb{R}^{n_i}$
$n_o \in \mathbb{N} =$ number of outputs, $\boldsymbol{y} \in \mathbb{R}^{n_o}$
$n_p \in \mathbb{N} =$ number of parameters
$n_s \in \mathbb{N} =$ number of states

$n_{sp} \in \mathbb{N}$ = number of B-spline basis and control points
$\wp \in \mathbb{N}$ = spline *degree*
$p_o \in \mathbb{N}$ = polynomial order

Dynamics

$\boldsymbol{C} \in \mathbb{R}^{n_s}$ = centrifugal, gyroscopic, and Coriolis terms
$\boldsymbol{H} \in \mathbb{R}^{n_s \times n_s}$ = kinematic mapping matrix relating rates of generalized coordinates to generalized velocities
$\boldsymbol{M} \in \mathbb{R}^{n_s \times n_s}$ = square inertia matrix
$\boldsymbol{N} \in \mathbb{R}^{n_s}$ = generalized gravitational and joint forces
$\boldsymbol{\mathcal{O}} \in \mathbb{R}^{n_o}$ = output operator
$\boldsymbol{q} \in \mathbb{R}^{n_d}$ = independent generalized coordinates
$\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ = rates and accelerations of generalized coordinates
$\boldsymbol{S} \in \mathbb{R}^{n_i \times n_d}$ = applied input selection matrix
$\boldsymbol{v}, \dot{\boldsymbol{v}} \in \mathbb{R}^{n_d}$ = generalized velocities and accelerations
$\boldsymbol{y} \in \mathbb{R}^{n_o}$ = system outputs
$\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ = uncertain parameters
$\boldsymbol{\tau} \in \mathbb{R}^{n_i}$ = input wrenches

Uncertainty Quantification

$A \in \mathbb{R}^{n_b \times n_{cp}}$ = collocation matrix
$w(\xi)$ = joint probability density function
${}_kX_i, X_i \in \mathbb{R}^{n_{cp}}$ = k^{th} intermediate variable of the ith state representing expanded quantity
${}_k\boldsymbol{\mu}, \mu \in \mathbb{R}^{n_{cp}}$ = kth collocation point
$\Psi \in \mathbb{R}^{n_b}$ = multidimensional basis terms
$\psi \in \mathbb{R}^{p_o+1}$ = single dimensional basis terms
Ω = random event sample space

Nonlinear Programming

$\mathbf{B}$ = B-spline curve
$\mathcal{C}$ = inequality constraints (typically bounding constraints)
$\mathfrak{D}_{i,j}$ = a signed minimum distance between two geometric bodies i and j
J = scalar objective function
$\min\limits_{\mathbf{x}}$ = optimization objective through the manipulation variables in $\mathbf{x}$
$\boldsymbol{P} = \{\boldsymbol{p}^i\}$ = B-spline control points where $i = 0, \ldots, n_{sp}$
$\boldsymbol{P}' = \left\{\boldsymbol{p}'^i\right\}$ = derived control points for velocity B-splines where $i = 0, \ldots, (n_{sp} - 1)$
$\boldsymbol{P}'' = \left\{\boldsymbol{p}''^i\right\}$ = derived control points for acceleration B-splines where $i = 0, \ldots, (n_{sp} - 2)$
t_f = final time of trajectories
z_i = scalarlization weights for the individual input wrench contributions
α_i = linear weight coefficient for defining uncertain constraints as a function of the std$[x]$, σ_x
$\beta^{i,\wp}$ = B-spline basis terms of *degree* $\wp$ and $i = 0, \ldots, n_{sp}$

References

[1] Park, J., 2007, "Optimal Motion Planning for Manipulator Arms Using Nonlinear Programming," *Industrial Robotics, Programming, Simulation and Applications*, J. K. Low, ed., Verlag, Croatia.
[2] Chong Jin, O., and Gilbert, E. G., 1996, "Growth Distances: New Measures for Object Separation and Penetration," IEEE Trans. Robotics Autom., **12**(6), pp. 888–903.
[3] Park, F., Bobrow, J., and Ploen, S., 1995, "A Lie Group Formulation of Robot Dynamics," Int. J. Robotics Res., **14**(6), pp. 609–618.
[4] Ploen, S., 1997, "Geometric Algorithms for the Dynamics and Control of Multibody Systems," Ph.D. thesis, University of California, Irvine, CA.
[5] Park, F. C., and Bobrow, J. E., 1994, "A Recursive Algorithm for Robot Dynamics Using Lie Groups," Proceedings of the 1994 IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1535–1540.
[6] Martin, B., and Bobrow, J., 1999, "Minimum-Effort Motions for Open-Chain Manipulators With Task-Dependent End-Effector Constraints," Int. J. Robotics Res., **18**(2), pp. 213–224.
[7] Sohl, G. A., and Bobrow, J. E., 2001, "A Recursive Multibody Dynamics and Sensitivity Algorithm for Branched Kinematic Chains," ASME J. Dyn. Syst. Measure. Control, **123**(3), pp. 391–399.
[8] Bobrow, J. E., and Sohl, G. A., 2002, "On the Reliable Computation of Optimal Motions for Underactuated Manipulators," Electron. J. Comp. Kinemat., **1**(1).
[9] Bobrow, J., Martin, B., Sohl, G., Wang, E., Park, F., and Kim, J., 2001, "Optimal Robot Motions for Physical Criteria," J. Robotic Syst., **18**(12), pp. 785–795.
[10] Sohl, G., 2000, "Optimal Dynamic Motion Planning for Underactuated Robots," Ph.D. thesis, University of California, Irvine, CA.
[11] Wang, C., Timoszyk, W., and Bobrow, J., 1999, "Weightlifting Motion Planning for a Puma 762 Robot," Proc. IEEE International Conference on Robotics and Automation, Vol. 1, pp. 480–485.
[12] Wang, C., 2001, "Dynamic Motion Planning for Robot Manipulators Using B-Splines," Ph.D. thesis, University of California, Irvine, CA.
[13] Junggon, K., Jonghyun, B., and Park, F. C., 1999, "Newton-Type Algorithms for Robot Motion Optimization," Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1842–1847.
[14] Bobrow, J. E., Park, F. C., and Sideris, A., 2006, *Fast Motions in Biomechanics and Robotics*, Springer, Berlin.
[15] Lee, S. H., Kim, J., Park, F. C., Kim, M., and Bobrow, J. E., 2005, "Newton-Type Algorithms for Dynamics-Based Robot Movement Optimization," IEEE Trans. Robotics, **21**(4), pp. 657–667.
[16] Xiang, Y., Arora, J., and Abdel-Malek, K., 2009, "Optimization-Based Motion Prediction of Mechanical Systems: Sensitivity Analysis," Struct. Multidisciplin. Optim., **37**(6), pp. 595–608.
[17] Xiang, Y., Chung, H., Mathai, A., Rahmatalla, S., Kim, J., Marler, T., Beck, S., Yang, J., Arora, J., and Abdel-Malek, K., 2007, "Optimization-Based Dynamic Human Walking Prediction," SAE Technical Paper 2007-01-2489.
[18] Chung, H., Xiang, Y., Mathai, A., Rahmatalla, S., Kim, J., Marler, T., Beck, S., Yang, J., Arora, J., and Abdel-Malek, K., 2007, "A Robust Formulation for Prediction of Human Running," DTIC, pp. 16–18.
[19] Xiang, Y., Arora, J. S., Rahmatalla, S., and Abdel-Malek, K., 2009, "Optimization-Based Dynamic Human Walking Prediction: One Step Formulation," Int. J. Numer. Methods Eng., **79**(6), pp. 667–695.
[20] Xiang, Y., Chung, H.-J., Kim, J., Bhatt, R., Rahmatalla, S., Yang, J., Marler, T., Arora, J., and Abdel-Malek, K., 2010, "Predictive Dynamics: An Optimization-Based Novel Approach for Human Motion Simulation," Struct. Multidisciplin. Optim., **41**(3), pp. 465–479.
[21] Xiang, Y., Arora, J., and Abdel-Malek, K., 2010, "Physics-Based Modeling and Simulation of Human Walking: A Review of Optimization-Based and Other Approaches," Struct. Multidisciplin. Optim., **42**(1), pp. 1–23.
[22] Xiang, Y., 2008, "Optimization-Based Dynamic Human Walking Prediction," Ph.D. thesis, University of Iowa, Iowa City, IA.
[23] Kim, H., Wang, Q., Rahmatalla, S., Swan, C., Arora, J., Abdel-Malek, K., and Assouline, J., 2008, "Dynamic Motion Planning of 3D Human Locomotion Using Gradient-Based Optimization," ASME J. Biomech. Eng., **130**(3), p. 031002.
[24] Diehl, M., Ferreau, H., and Haverbeke, N., 2009, "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," Nonlinear Model Predictive Control, **384**, pp. 391–417.
[25] Biegler, L. T., 2003, "Optimization of ODE/DAE Constrained Models," http://www.ima.umn.edu/talks/workshops/1-8.2003/biegler/IMA_Tutorial.pdf
[26] Lavalle, S., 2006, *Planning Algorithms*, Cambridge University Press, New York.
[27] Barraquand, J., and Ferbach, P., 1995, "Motion Planning With Uncertainty: The Information Space Approach," Int. Conf. Robotics Autom., **2**, pp. 1341–1348.
[28] Park, W., Liu, Y., Zhou, Y., Moses, M., and Chirikjian, G., 2008, "Kinematic State Estimation and Motion Planning for Stochastic Nonholonomic Systems Using the Exponential Map," Robotica, **26**(04), pp. 419–434.
[29] Erdmann, M., 1984, "On Motion Planning With Uncertainty," Masters thesis, Massachusetts Institute of Technology, Boston, MA.
[30] Kewlani, G., Ishigami, G., and Iagnemma, K., 2009, "Stochastic Mobility-Based Path Planning in Uncertain Environments," IROS 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1183–1189.
[31] Hays, J., Sandu, A., Sandu, C., and Hong, D., 2011, "Motion Planning of Uncertain Fully-Actuated Dynamical Systems—An Inverse Dynamics Formulation," ASME IDETC/CIE Conference, Washington, DC, Aug. 28–30, pp. 355–360.
[32] Hays, J., Sandu, A., Sandu, C., and Hong, D., 2011, "Motion Planning of Uncertain Fully-Actuated Dynamical Systems—a Forward Dynamics Formulation," ASME IDETC/CIE Conference, Washington, DC.
[33] Hays, J., Sandu, A., Sandu, C., and Hong, D., 2011, "Motion Planning of Uncertain Under-Actuated Dynamical Systems—a Hybrid Dynamics Formulation," Proceedings of the ASME 2011 International Mechanical Engineering Congress and Exposition, Denver, CO, Nov. 11–17, pp. 729–736.
[34] Wiener, N., 1938, "The Homogeneous Chaos," Am. J. Math., **60**(4), pp. 897–936.
[35] Xiu, D., and Karniadakis, G., 2002, "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," SIAM J. Sci. Comput., **24**(2), pp. 619–644.
[36] Xiu, D., 2009, "Fast Numerical Methods for Stochastic Computations: A Review," Commun. Comput. Phys., **5**(2–4), pp. 242–272.
[37] Xiu, D., and Hesthaven, J. S., 2005, "High-Order Collocation Methods for Differential Equations With Random Inputs," SIAM J. Sci. Comput., **27**(3), pp. 1118–1139.
[38] Xiu, D., 2007, "Efficient Collocational Approach for Parametric Uncertainty Analysis," Commun. Comput. Phys., **2**(2), pp. 293–309.

[39] Sandu, A., Sandu, C., and Ahmadian, M., 2006, "Modeling Multibody Systems With Uncertainties. Part I: Theoretical and Computational Aspects," Multibody Syst. Dyn., **15**(4), pp. 369–391.
[40] Cheng, H., and Sandu, A., 2009, "Efficient Uncertainty Quantification With the Polynomial Chaos Method for Stiff Systems," Math. Comput. Simul., **79**(11), pp. 3278–3295.
[41] Wan, X., and Karniadakis, G., 2005, "An Adaptive Multi-Element Generalized Polynomial Chaos Method for Stochastic Differential Equations," J. Comput. Phys., **209**(2), pp. 617–642.
[42] Wan, X., and Karniadakis, G., 2006, "Adaptive Numerical Solutions of Stochastic Differential Equations," Comput. Math. Appl., pp. 561–573.
[43] Wan, X., and Karniadakis, G., 2006, "Beyond Wiener–Askey Expansions: Handling Arbitrary Pdfs," J. Sci. Comput., **27**(1), pp. 455–464.
[44] Wan, X., and Karniadakis, G., 2007, "Multi-Element Generalized Polynomial Chaos for Arbitrary Probability Measures," SIAM J. Sci. Comput., **28**(3), pp. 901–928.
[45] Foo, J., Wan, X., and Karniadakis, G., 2008, "The Multi-Element Probabilistic Collocation Method: Error Analysis and Simulation," J. Comput. Phys, **227**(22), pp. 9572–9595.
[46] Foo, J., and Karniadakis, G. E., 2010, "Multi-Element Probabilistic Collocation Method in High Dimensions," J. Comput. Phys., **229**(5), pp. 1536–1557.
[47] Sandu, C., Sandu, A., and Ahmadian, M., 2006, "Modeling Multibody Systems With Uncertainties. Part II: Numerical Applications," Multibody Syst. Dyn., **15**(3), pp. 241–262.
[48] Cheng, H., and Sandu, A., 2007, "Numerical Study of Uncertainty Quantification Techniques for Implicit Stiff Systems," Proceedings of the 45th ACM Southeast Conference, Winston-Salem, NC, pp. 367–372.
[49] Cheng, H., and Sandu, A., 2009, "Uncertainty Quantification in 3D Air Quality Models Using Polynomial Chaoses," Environ. Model. Software, **24**(8), pp. 917–925.
[50] Cheng, H., and Sandu, A., 2009, "Uncertainty Apportionment for Air Quality Forecast Models," Proceedings of 24th Annual ACM Symposium on Applied Computing (SAC-2009), Computational Sciences Track, Honolulu, HI, pp. 956–960.
[51] Cheng, H., and Sandu, A., 2010, "Collocation Least-Squares Polynomial Chaos Method," Proceedings of the 2010 Spring Simulation Multiconference (SpringSim'10), High Performance Computing Symposium (HPC-2010), Orlando, FL, p. 80.
[52] Blanchard, E., 2010, "Polynomial Chaos Approaches to Parameter Estimation and Control Design for Mechanical Systems With Uncertain Parameters," Ph.D. thesis, Virginia Tech, Blacksburg, VA.
[53] Blanchard, E., Sandu, A., and Sandu, C., 2007, "Parameter Estimation Method Using an Extended Kalman Filter," Joint North America, Asia-Pacific ISTVS Conference, Fairbanks, AK, pp. 23–26.
[54] Blanchard, E., Sandu, A., and Sandu, C., 2009, "Parameter Estimation for Mechanical Systems Via an Explicit Representation of Uncertainty," Eng. Comput., **26**(5), pp. 541–569.
[55] Blanchard, E., Sandu, A., and Sandu, C., 2010, "Polynomial Chaos-Based Parameter Estimation Methods Applied to a Vehicle System," Proc. Inst. Mech. Eng., Part K: J. Multi-body Dyn., **224**(1), pp. 59–81.
[56] Blanchard, E., Sandu, A., and Sandu, C., 2010, "Polynomial Chaos Based Method for the LQR Problem With Uncertain Parameters in the Formulation," Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Canada, Aug. 15–18, pp. 315–324.
[57] Blanchard, E., Sandu, C., and Sandu, A., 2007, "A Polynomial-Chaos-Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems," Proceedings of the ASME IDETC 2007, 9th International Conference on Advanced Vehicle and Tire Technology, Las Vegas, NV, pp. 4–7.
[58] Blanchard, E., Sandu, C., and Sandu, A., 2009, "Comparison Between a Polynomial-Chaos-Based Bayesian Approach and a Polynomial-Chaos-Based EKF Approach for Parameter Estimation With Application to Vehicle Dynamics," Proceedings of the ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 11th International Conference on Advanced Vehicle and Tire Technology, San Diego, CA, Aug. 30–Sept. 2, ASME Paper No. DETC2009-86402, pp. 893–904.
[59] Blanchard, E., and Sandu, D., 2007, "A Polynomial Chaos Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems-Part 2: Applications to Vehicle Systems," Technical Report No. TR-07-39, Virginia Tech, Blacksburg, VA.
[60] Blanchard, E., and Sandu, D., 2007, "A Polynomial Chaos Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems–Part 1: Theoretical Approach," Technical Report No. TR-07-38, Virginia Tech, Blacksburg, VA.
[61] Blanchard, E. D., Sandu, A., and Sandu, C., 2010, "A Polynomial Chaos-Based Kalman Filter Approach for Parameter Estimation of Mechanical Systems," ASME J. Dyn. Syst. Measure. Control, **132**(6), p. 061404.
[62] Pence, B., Hays, J., Fathy, H., Sandu, C., and Stein, J., 2013, "Vehicle Sprung Mass Estimation for Rough Terrain," Int. J. Veh. Design **61**(1), pp. 3–36.
[63] Pence, B. L., Fathy, H. K., and Stein, J. L., 2009, "A Base-Excitation Approach to Polynomial Chaos-Based Estimation of Sprung Mass for Off-Road Vehicles," ASME Dynamic Systems and Control Conference, Hollywood, CA, Oct. 12–14, ASME Paper No. DSCC2009-2641, pp. 857–864.
[64] Pence, B. L., Fathy, H. K., and Stein, J. L., 2010, "Recursive Bayesian Parameter Estimation Using Polynomial Chaos Theory," Automatica, **47**(11), pp. 2420–2424.
[65] Pence, B. L., Fathy, H. K., and Stein, J. L., 2010, "An Integrated Cost/Maximum Likelihood Approach to Recursive Polynomial Chaos Parameter Estimation," American Control Conference (ACC), June 30–July 2, pp. 2144–2151.
[66] Southward, S., 2007, "Real-Time Parameter ID Using Polynomial Chaos Expansions," ASME Conf. Proc. 43033, pp. 1167–1173.
[67] Shimp III, S., 2008, "Vehicle Sprung Mass Parameter Estimation Using an Adaptive Polynomial-Chaos Method," Masters thesis, Virginia Tech, Blacksburg, VA.
[68] Marzouk, Y., and Xiu, D., 2009, "A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems," Commun. Comput. Phys., **6**, pp. 826–847.
[69] Marzouk, Y. M., Najm, H. N., and Rahn, L. A., 2007, "Stochastic Spectral Methods for Efficient Bayesian Solution of Inverse Problems," J. Comput. Phys., **224**(2), pp. 560–586.
[70] Price, D., 2008, "Estimation of Uncertain Vehicle Center of Gravity Using Polynomial Chaos Expansions," Masters thesis, Virginia Tech, Blackburg, VA.
[71] Smith, A., Monti, A., and Ponci, F., 2007, "Indirect Measurements Via a Polynomial Chaos Observer," IEEE Trans. Instrum. Measure., **56**(3), pp. 743–752.
[72] Li, J., and Xiu, D., 2009, "A Generalized Polynomial Chaos Based Ensemble Kalman Filter With High Accuracy," J. Comput. Phys., **228**(15), pp. 5454–5469.
[73] Saad, G., Ghanem, R., and Masri, S., 2007, "Robust System Identification of Strongly Non-Linear Dynamics Using a Polynomial Chaos Based Sequential Data Assimilation Technique," **6**, pp. 6005–6013.
[74] Templeton, B., 2009, "A Polynomial Chaos Approach to Control Design," Ph.D. thesis, Virginia Tech, Blacksburg, VA.
[75] Smith, A., Monti, A., and Ponci, F., 2006, "Robust Controller Using Polynomial Chaos Theory," Industry Applications Conference, Tampa, FL, Oct. 8–12, pp. 2511–2517.
[76] Prempraneerach, P., Hover, F., Triantafyllou, M., and Karniadakis, G., 2010, "Uncertainty Quantification in Simulations of Power Systems: Multi-Element Polynomial Chaos Methods," Reliability Eng. Syst. Safety, **95**, pp. 632–646.
[77] Kewlani, G., and Iagnemma, K., 2009, "A Multi-Element Generalized Polynomial Chaos Approach to Analysis of Mobile Robot Dynamics Under Uncertainty," pp. 1177–1182.
[78] Greenwood, D., 2003, *Advanced Dynamics*, Cambridge University Press, Cambridge, UK.
[79] Murray, R., Li, Z., Sastry, S., and Sastry, S., 1994, *A Mathematical Introduction to Robotic Manipulation*, CRC, Boca Raton, FL.
[80] Nikravesh, P. E., 2004, *Product Engineering, an Overview of Several Formulations for Multibody Dynamics*, Springer, Berlin.
[81] Haug, E. J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems*. Vol. 1: Basic Methods, Allyn & Bacon, Boston, MA.
[82] Piegl, L. A., and Tiller, W., 1997, *The Nurbs Book*, Springer, Berlin.