

**Computer Science Technical Report**  
**TR-11-04**  
**April 20, 2011**

**Joe Hays, Adrian Sandu, Corina Sandu, Dennis Hong**

*“Motion Planning of Uncertain Ordinary Differential  
Equation Systems”*

**Center for Vehicle Systems and Safety  
Computer Science Department & Department of Mechanical  
Engineering  
Virginia Polytechnic Institute and State University  
Blacksburg, VA 24061  
Phone: (540)-231-2193  
Fax: (540)-231-9218  
Email: sandu@cs.vt.edu  
Web: <http://www.eprints.cs.vt.edu>**



# Motion Planning of Uncertain Ordinary Differential Equation Systems

**Joe Hays**

Mechanical Engineering  
Virginia Tech  
Blacksburg, VA 24061

**Adrian Sandu**

Computational Science Laboratory  
Computer Science Department  
Virginia Tech  
Blacksburg, VA 24061

**Corina Sandu**

Advanced Vehicle Dynamics Laboratory  
Mechanical Engineering  
Virginia Tech  
Blacksburg, VA 24061

**Dennis Hong**

Robotics and Mechanisms Laboratory  
Mechanical Engineering  
Virginia Tech  
Blacksburg, VA 24061

## Abstract

*This work presents a novel motion planning framework, rooted in nonlinear programming theory, that treats uncertain fully and under-actuated 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's 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 under-actuated systems, prescribes deterministic actuator inputs which yield uncertain state trajectories. The inverse dynamics formulation is the dual to the forward dynamic, 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 under-actuated 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 which yield uncertain unactuated states and 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.*

**Keywords:** Motion Planning, Trajectory Planning, Optimization, Nonlinear Programming, Multibody Dynamics, Uncertainty Quantification

## List of Variables (Nomenclature)

Independent variables	
$t$	Time
$\omega$	Random event
General	
$x, X$	Non-bolded variables generally indicate a scalar quantity
$\mathbf{x}, \mathbf{X}$	Bolded lower case variables are vectors, upper case variables are matrices
$\xi$	Random variable
$x_i$	Bottom right index <i>generally</i> indicates a state (with occasional exceptions).
$x^j$	Top right index generally indicates a stochastic coefficient, or mode.
${}_k x$	Bottom left index generally associates $x$ to a specific collocation point.
${}^a x, {}^u x$	Top left annotations indicate if a given variable is <i>actuated</i> or <i>unactuated</i> .
${}^u{}_k x_i^j$	The four major variable annotations
$( )^T$	Transpose
$( )_{\mathbf{q}}, \frac{\partial}{\partial \mathbf{q}}$	Partial derivative notations
$( )^{-1}, ( )^{\#}$	Matrix inverse and pseudo inverse
$x, \bar{x}$	Lower and upper bounds on $x$
$E[x], \mu_x$	Expected value, or mean, of $x$
$Var[x], \sigma_x^2$	Variance of $x$
$std[x], \sigma_x$	Standard Deviation of $x$
$inf(x), sup(x)$	Infimum and supremum of $x$
Indexes & dimensions	
$n_d \in \mathbb{N}$	Number of degrees-of-freedom (DOF)
$n_s \in \mathbb{N}$	Number of states
$n_p \in \mathbb{N}$	Number of parameters
$n_i \in \mathbb{N}$	Number of input wrenches, $\boldsymbol{\tau} \in \mathbb{R}^{n_i}$
$n_o \in \mathbb{N}$	Number of outputs, $\mathbf{y} \in \mathbb{R}^{n_o}$
$p_o \in \mathbb{N}$	Polynomial order
$n_b \in \mathbb{N}$	Number of multidimensional basis terms
$n_{cp} \in \mathbb{N}$	Number of collocation points
$n_{sp} \in \mathbb{N}$	Number of B-Spline basis and control points
$m \in \mathbb{N}$	Number of B-Spline knots
$p \in \mathbb{N}$	Spline <i>degree</i>
$n_{dim} \in \mathbb{N}$	Number of dimensions of the B-Spline (e.g., $n_d$ or $n_i$ )
Dynamics	
$\mathbf{q} \in \mathbb{R}^{n_d}$	Independent generalized coordinates
$\dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Rates and accelerations of generalized coordinates
$\mathbf{v} \in \mathbb{R}^{n_d}$	Generalized velocities
$\dot{\mathbf{v}}$	Generalized accelerations
$\mathbf{q}(\mathbf{0}) = \mathbf{q}_0, \mathbf{v}(\mathbf{0}) = \mathbf{v}_0$	Initial conditions
$\mathbf{H} \in \mathbb{R}^{n_s \times n_s}$	Kinematic mapping matrix relating rates of generalized coordinates to generalized velocities
$\boldsymbol{\theta} \in \mathbb{R}^{n_p}$	Uncertain parameters
$\boldsymbol{\tau} \in \mathbb{R}^{n_i}$	Input wrenches
$\mathbf{M} \in \mathbb{R}^{n_s \times n_s}$	Square inertia matrix
$\mathbf{C} \in \mathbb{R}^{n_s}$	Centrifugal, gyroscopic and Coriolis terms
$\mathbf{N} \in \mathbb{R}^{n_s}$	Generalized gravitational and joint forces
$\mathcal{F}$	Differential operator
$\mathcal{G}$	Under-actuated differential operator
$\mathbf{y} \in \mathbb{R}^{n_o}$	System outputs
$\mathcal{O} \in \mathbb{R}^{n_o}$	Output operator
Uncertainty Quantification	
$\Omega$	Random event sample space
$w(\xi)$	Joint probability density function
$\psi \in \mathbb{R}^{p_o+1}$	Single dimensional basis terms
$\Psi \in \mathbb{R}^{n_b}$	Multidimensional basis terms

$k\boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^{n_{cp}}$	$K^{\text{th}}$ collocation point
$kX_i, X_i \in \mathbb{R}^{n_{cp}}$	$K^{\text{th}}$ intermediate variable of the $i^{\text{th}}$ state representing expanded quantity
$A \in \mathbb{R}^{n_b \times n_{cp}}$	Collocation matrix
<b>Nonlinear Programming</b>	
$\min_{\mathbf{x}}$	Optimization objective through manipulation of $\mathbf{x}$
$\mathbf{x}$	List of manipulated variables
$J$	Scalar objective function
$z_i$	Scalarization weights for the individual input wrench contributions
$t_f$	Final time of trajectories
$\mathcal{C}$	Inequality constraints (typically bounding constraints)
$\mathbf{B}$	B-Spline curve
$\beta^{i,p}$	B-Spline basis terms of <i>degree</i> $p$ and $i = 1 \dots n_{sp}$
$\mathbf{P} = \{\mathbf{p}^i\}$	B-Spline control points where $i = 1 \dots n_{sp}$
$\mathbf{P}' = \{\mathbf{p}'^i\}$	Derived control points for velocity B-Splines where $i = 1 \dots n_{sp}$
$\mathbf{P}'' = \{\mathbf{p}''^i\}$	Derived control points for acceleration B-Splines where $i = 1 \dots n_{sp}$
$\mathcal{D}_{i,j}$	A signed minimum distance between two geometric bodies $i$ and $j$

# 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 down-stream 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 (see Figure 1.1); medical robots are now being designed to aid physicians in surgery; and autonomous vehicles are taking on more and more tasks in military, municipality, and even domestic operations.



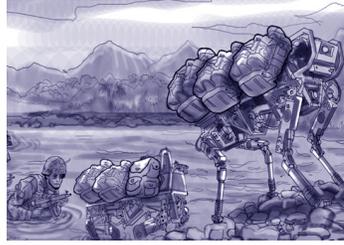
<http://img.directindustry.com>



<http://www.drives.co.uk>

**Figure 1.1—Industrial robots are example applications of engineered systems whose robustness and performance can be improved by the proper treatment of uncertainty during the motion planning process.**

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 (UUV's) are outfitted with articulated accessories to perform various tasks. These systems are planned to aid in diverse operations including Improvised Incendiary Device (IID) detection and disarmament, material and equipment handling and convoy, search and rescue. Three show-case examples include: Boston Dynamics' BigDog and next generation LS3 robots, who aid in the convoy of soldier equipment with an unknown weight in harsh rugged terrain; Vecna's BEAR robot aids in the retrieval of wounded soldiers of varying size and weight also in uncertain 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 achievable distances of a given convoy operation (see Figure 1.2).



<http://www.bostondynamics.com>



<http://www.xconomy.com>

**Figure 1.2—Autonomous robotic systems illustrate multibody dynamic systems that operate within uncertain environments and payloads.**

These are a few showcase examples of specific dynamic system applications that would benefit if uncertainty was accounted for during the motion planning process.

Another noteworthy application is in the area of studying and analyzing human performance measures. For example, TARDEC [1-4] is actively investigating the effects of protective clothing and routine tasks of soldiers, such as: crawling, walking, running, lifting, calisthenics and other human physiology aspects. The studies aim to quantify muscle stress/fatigue, metabolic rates, and required strength of tasks expected of soldiers (see Figure 1.3); where techniques typically involve dynamic analysis and various optimal motion planning formulations of virtual and physical humanoids. Recently the automotive manufacturing sector is using similar analyses to ensure the designed assembly lines are safe for their employees [5]. Literature to date has shown that these efforts currently assume a deterministic dynamic system model. Clearly soldiers and assembly line workers operate with uncertain payloads and tasks, therefore, the ability to quantify and account for these uncertainties would enrich these human performance studies.



<http://www.ccad.uiowa.edu/vsr/>

**Figure 1.3—Optimization based human performance studies such as quantifying muscle stress/fatigue, metabolic rates, and required strength can be enriched with the proper treatment of uncertainty.**

## 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 [6], Park presents a nonlinear programming approach to motion planning for robotic manipulator arms described by 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 [7].

Sohl, Martin, and Bobrow presented a series of papers that document their excellent work in the area of optimal manipulator motions. At the heart of their work is the use of a novel geometric formulation of robot dynamics based on the differential geometry principles of Lie Groups and Lie Algebras [8-10]. The approach provides a few critical properties that streamline the optimal motion planning problem; first, the geometric dynamics formulation has an equivalent recursive formulation that provides  $O(n)$  computational complexity; second, use of the Product-of-Exponentials (POE) in the formulation provides a straight-forward approach to calculating the gradient of the optimal motion planning objective function. Access to an exact analytic gradient improves the nonlinear programming solve by helping avoid premature convergence or excessive searching for the frequently ill-conditioned motion planning problems. In [11], Martin and Bobrow present a minimum effort formulation for open chain manipulators based on the recursive geometric dynamics. A detailed presentation for the recursive calculation of the analytic objective function gradient is a major contribution of this work. They also use cubic B-Splines to provide a finite dimensional search space. In [12], Sohl and Bobrow extend the work to address branched kinematic chains; in [13-15] they again extend the work to address under-actuated manipulators; and in [16, 17] 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 for the constrained optimization; however, in [18], a Newton type optimization algorithm is developed that reuses the analytic gradient and hessian information from the geometric dynamics. In [19], Bobrow, Park, and Sideris, further extend the work to solve infinite-dimensional problems using a sequence of linear-quadratic optimal control sub-problems and cover minimum energy, control effort, jerk and time. Finally, in [20], Lee et. al. extend 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, Abdel-Malek et. al. [2-4, 21-25] 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 over-actuated 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 [26, 27].) Additional contributions of Xiang's work include human walking specific constraint formulations.

In [28], Park and Park present a convex motion planning algorithm that determines a stable motion plan that approximates a reference motion plan for a humanoid robotic system. The use case stems from applying measured joint trajectories from a human and applying them to a humanoid robot; this generally results in an unstable reference trajectory for the robot. However, Park and Park present a second-order cone formulated motion planning problem that determines a stable motion plan yet still approximates the reference trajectory in a least-squares sense. Similar work was presented in [29], where reference motion plans are refined online through use of a recursive *forward dynamics* based optimization framework with analytic derivatives. The resulting motion plan is determined in the *joint space* versus the *wrench space*.

Lim et. al. present an interesting extension to the optimal motion planning problem in [30], where motion primitives are extracted from an ensemble of optimal motions determined through repeated optimizations of a perturbed walking surface. The technique is applied to the novel tripod robot STriDER. The primitives are determined by extracting principle components from the ensemble of optimal motion plans over varying heights of the walking surface. Once determined, the motion primitives provide a fast reference motion plan for online use. Unlike the previously referenced papers, Lim's work used Power Series to parameterize the infinite search space. The design sought for a minimum effort gait. Hays et al. have investigated the co-design of STriDER's motion plan and mechanical properties in [31].

### 1.2.2 Sample-Based Motion Planning

Unlike nonlinear programming optimization searches, *sample-based motion planning* techniques randomly sample a design space in search of a feasible motion plan. Some predominant examples of these techniques include: Rapidly-exploring Random Trees (RRTs) [32, 33], probabilistic Roadmaps (PRMs) [32, 33], and the relatively new Rapidly-exploring Random Graph (RRGs) [34].

### 1.2.3 Motion Planning of Uncertain Systems

Very little research has been performed in the area motion planning of uncertain systems. LaValle treats sensor uncertainty with RRTs in [32]. Barraquand addresses both actuator and sensor uncertainty in a stochastic dynamic programming (DP) framework but this work only addresses the kinematics of the system [35]. Park also presents a kinematic only motion planning solution for systems with sensor and actuator uncertainties based on the Fokker-Planck equation [36]. Erdmann's early work on the *back-projection* method also only addressed sensor and actuator noise and was limited to first-order dynamic models [37].

In [38], 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). 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, versus a feasible, motion planning for uncertain dynamical systems [39-41].

### 1.2.4 Monte Carlo Uncertainty Quantification

The Monte Carlo (MC) method is considered the most robust method of uncertainty quantification. The method is quite simple; the probability space of the system is randomly sampled  $n$  times and statistical measures are determined from the ensemble [42]. MC provides a consistent error convergence rate independent of the number of uncertainties. However, the convergence rate of  $1/\sqrt{n}$  is relatively slow.

Alternatively, quasi-Monte Carlo (QMC) methods deterministically sample the probability space with *low-discrepancy sequences* (LDS). QMC is reported to show improved constant convergence,  $(\log n)^d/n$ , for relatively low dimensional problems when compared to MC [43, 44]; where  $d$  is the number of dimensions.

### 1.2.5 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 the 1930's where he introduced the idea of *homogenous chaoses* [45]. 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 [46, 47]; where the Galerkin Projection Method (GPM) was initially used. In [47-49], Xiu showed an initial collocation method based on Lagrange interpolation. A number of Collocation point selection methods were also show including tensor products and Smolyak sparse grids.

In [50], 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 the LSCM maintains the exponential convergence of GPM yet was superior in computational speed in [51]; where the Hammersley LDS data set was the preferred method in selecting collocation points. Cheng and Sandu also presented a modified time stepping mechanism where an approximate Jacobian was used when solving stiff systems.

### 1.2.6 Multi-Element gPC

The accuracy of gPC deteriorates over time in long simulations and is dependent on continuity of the system. In an effort to address these two concerns, Wan and Karniadakis developed multi-element gPC (MEgPC) [52, 53]. This method discretizes the probability space into non-overlapping 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 [54, 55]. Foo developed a collocation-based MEgPC in [56] and further extended the method to support higher dimensions using ANOVA methods in [57].

As an alternative to MEgPC, Witteveen and Iaccarino developed a similar multi-element 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.

### 1.2.7 Recent Applications of gPC/MEgPC

The origins of gPC come from thermal/fluid applications; however, its adoption in other areas continues to expand. Sandu and coworkers introduced its application to multibody dynamical systems in [50, 51, 58-62]. Significant work has been done applying it as a foundational element in parameter [46-49, 63-81] and state estimation [82, 83], as well as system identification [84]. Relatively recent work has applied gPC to both classical and optimal control system design [63, 85, 86]. Also, MEgPC has been used applied to uncertainty quantification in power systems [87] and mobile robots [88].

## 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 under-actuated 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 result if it's 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 through forward, inverse, and hybrid dynamics formulations. The forward dynamics formulation, applicable to both fully and under-actuated systems, prescribes deterministic actuator inputs which yield uncertain state trajectories. The inverse dynamics formulation, however, is the dual to the forward dynamics formulation and is only applicable to fully-actuated systems; it has prescribed deterministic state trajectories which yield 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 as applicable to under-actuated 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 which yield uncertain unactuated states and 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's 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 Section 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 under-actuated dynamical systems based on the uncertain forward, inverse, and hybrid dynamics formulations. Section 6 illustrates the strengths of the new framework through a series of case-studies. Concluding remarks are presented in Section 7.

## 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 [89, 90],

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

where  $\mathbf{q}(t) \in \mathbb{R}^{n_d}$  are independent generalized coordinates equal in number to the number of degrees of freedom,  $n_d$ ;  $\mathbf{v}(t) \in \mathbb{R}^{n_d}$  the generalized velocities and—using Newton’s *dot* notation— $\dot{\mathbf{v}}(t)$  contains their time derivatives;  $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_p}$  includes system parameters of interest;  $\mathbf{M}(\mathbf{q}(t), \boldsymbol{\theta}(t)) \in \mathbb{R}^{n_d \times n_d}$  is the square inertia matrix;  $\mathbf{C}(\mathbf{q}(t), \mathbf{v}(t), \boldsymbol{\theta}(t)) \in \mathbb{R}^{n_d \times n_d}$  includes centrifugal, gyroscopic and Coriolis effects;  $\mathbf{N}(\mathbf{q}(t), \mathbf{v}(t), \boldsymbol{\theta}(t)) \in \mathbb{R}^{n_d}$  the generalized gravitational and joint forces; and  $\boldsymbol{\tau}(t) \in \mathbb{R}^{n_i}$  are the  $i$  applied wrenches. (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,

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

where  $\mathbf{H}(\mathbf{q}, \boldsymbol{\theta})$  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.) [41, 91, 92]. However, if (1) is formulated with independent generalized coordinates and the system has a fixed base, as in [39, 40], then (2) becomes  $\dot{\mathbf{q}} = \mathbf{v}$ .

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

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

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

## 3 DETERMINISTIC MOTION PLANNING OF UNDER-ACTUATED 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 under-actuated systems presents multiple methodologies for formulating the governing dynamics. The *forward dynamics* formulation, applicable to both fully and under-actuated systems, prescribes actuator inputs which 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 which 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 under-actuated 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,  $\mathbf{q} = \{^a\mathbf{q}, ^u\mathbf{q}\}$  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).

**Table 1—Knowns vs Unknowns Dynamic Properties**

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

Regardless of which dynamics formulation is selected, a common motion planning practice is to approximate infinite dimensional *known* trajectories by a finite dimensional parameterization [15]. This paper parameterizes all *known* trajectories with B-Splines. For example, the parameterization of  $\mathbf{q}$  takes the form,

$$\mathbf{q}(\mathbf{P}, u) = \sum_{i=0}^{n_{sp}} \beta^{i, \mathcal{P}-1}(u) \mathbf{p}^i \quad (4)$$

and a similar expansion is given for  $\boldsymbol{\tau}(\mathbf{P}, u)$ . There are  $(n_{sp} + 1)$  control points  $\mathbf{P} = \{\mathbf{p}^0, \dots, \mathbf{p}^{n_{sp}}\} \in \mathbb{R}^{n_{sp}+1} \times \mathbb{R}^{n_{dim}}$  with  $\mathbf{p}^i \in \mathbb{R}^{n_{dim}}$ , where  $\mathbf{p}^{i,j}$  is the  $j^{th}$  element of the  $i^{th}$  control point;  $m + 1$  non-decreasing knots  $u^0 \leq \dots \leq u^m \in \mathbb{R}$ ; and  $(n_{sp} + 1)$  basis  $\beta^{i, \mathcal{P}}(u)$  of degree of  $\mathcal{P}$ ; and the relation  $m = n_{sp} + \mathcal{P} + 1$  must be maintained.

Basis functions,  $\beta^{i, \mathcal{P}}(u)$ , can be created recursively by the *Cox-de Boor recursion formula*.

$$\beta^{i,0}(u) = \begin{cases} 1 & \text{if } u^i \leq u < u^{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\beta^{i,\mathcal{P}}(u) = \frac{u - u^i}{u^{i+\mathcal{P}} - u^i} \beta^{i,\mathcal{P}-1}(u) + \frac{u^{i+\mathcal{P}+1} - u}{u^{i+\mathcal{P}+1} - u^{i+1}} \beta^{i+1,\mathcal{P}-1}(u)$$

Also, a *clamped* B-spline has  $(\mathcal{P} + 1)$  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 at the first and last control points. Meaning,  $\boldsymbol{\tau}(\mathbf{P}, u^0) = \mathbf{p}^0$  and  $\boldsymbol{\tau}(\mathbf{P}, u^m) = \mathbf{p}^{n_{sp}}$ . 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  $\mathbf{q}'(\mathbf{P}, u) = \frac{\partial \mathbf{q}(u)}{\partial u}$  represent the first derivative of  $\mathbf{q}(\mathbf{P}, u)$ . With a slight abuse of Lagrange's derivative notation, let the control points for  $\mathbf{q}'(\mathbf{P}, u)$  be defined as  $\mathbf{P}' = \{\mathbf{p}'^0, \dots, \mathbf{p}'^{n_{sp}-1}\}$ . Unlike  $\mathbf{P}$ , the values of  $\mathbf{P}'$  are predetermined through the following recursive relation,

$$\mathbf{p}'^i = \frac{\mathcal{P}}{u^{i+\mathcal{P}+1} - u^{i+1}} (\mathbf{p}^{i+1} - \mathbf{p}^i) \quad (6)$$

which gives the  $n_{sp} - 1$  inherited control points; or,  $\mathbf{P}' \in \mathbb{R}^{n_{sp}-1} \times \mathbb{R}^{n_{dim}}$ . The corresponding  $n_{sp} - 1$  basis functions,  $\beta^{i,\mathcal{P}-1}(u)$ , are of degree  $\mathcal{P} - 1$  and are also calculated using (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,  $\mathbf{u}'$ , is updated by removing the first and last knot from the original knot vector,  $\mathbf{u}$ ,

$$\mathbf{u}' = \{u^1 \leq \dots \leq u^{m-1}\} \subset \mathbf{u}. \quad (7)$$

These recursive relations for control points, basis, and knot vectors also apply for higher-order derivatives. Therefore, by defining  $\mathbf{P}$  for  $\mathbf{q}(\mathbf{P}, u)$ , all of its derivatives supported by the original degree  $\mathcal{P}$ , control points, and knots, are automatically defined [93].

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

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

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

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 (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  $\beta^{i,\mathcal{P}}(u) = \beta^{i,\mathcal{P}}(t)$ . Therefore, the curves  $\mathbf{q}(\mathbf{P}, u) = \mathbf{q}(\mathbf{P}, t)$  and  $\boldsymbol{\tau}(\mathbf{P}, u) = \boldsymbol{\tau}(\mathbf{P}, t)$  are defined from  $[t_0, t_f)$ .

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

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

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

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

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

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

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

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

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

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

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  $\{{}^a \mathbf{q}(\mathbf{P}), {}^a \mathbf{v}(\mathbf{P}), {}^a \dot{\mathbf{v}}(\mathbf{P})\}$  are specified. In such a formulation (15) reduces to,

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

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

**Forward Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{\mathbf{P}\}} J \\ \text{s. t. } & \mathcal{F}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\theta}) = \boldsymbol{\tau}(\mathbf{P}) \\ & \dot{\mathbf{q}} = \mathbf{H}(\mathbf{q}, \boldsymbol{\theta})\mathbf{v} \\ & \mathbf{y} = \mathcal{O}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\theta}) \\ & \mathcal{C}(\mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{0} \\ & \mathbf{q}(0) = \mathbf{q}_0 \\ & \dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0, \\ & \mathbf{q}(t_f) = \mathbf{q}_{t_f} \\ & \dot{\mathbf{q}}(t_f) = \dot{\mathbf{q}}_{t_f} \end{aligned} \quad (17)$$

**Inverse Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{\mathbf{P}\}} J \\ \text{s. t. } & \mathbf{v}(\mathbf{P}') = (\mathbf{H}(\mathbf{q}(\mathbf{P}), \boldsymbol{\theta}))^{-1} \dot{\mathbf{q}}(\mathbf{P}') \\ & \dot{\mathbf{v}}(\mathbf{P}'') = (\mathbf{H}(\mathbf{q}(\mathbf{P}), \boldsymbol{\theta}))^{-1} \left( \ddot{\mathbf{q}}(\mathbf{P}'') - \mathbf{v}(\mathbf{P}') \left( \frac{\partial \mathbf{H}}{\partial t} + \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial t} \right) \right) \\ & \boldsymbol{\tau} = \mathcal{F}(\mathbf{q}(\mathbf{P}), \mathbf{v}(\mathbf{P}'), \dot{\mathbf{v}}(\mathbf{P}''), \boldsymbol{\theta}) \\ & \mathbf{y} = \mathcal{O}(\mathbf{q}(\mathbf{P}), \dot{\mathbf{q}}(\mathbf{P}'), \boldsymbol{\theta}) \\ & \mathcal{C}(\mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{0} \\ & \mathbf{q}(0) = \mathbf{P}^0 = \mathbf{q}_0 \\ & \dot{\mathbf{q}}(0) = \mathbf{P}'^0 = \dot{\mathbf{q}}_0 \\ & \mathbf{q}(t_f) = \mathbf{P}^{n_{sp}} = \mathbf{q}_{t_f} \\ & \dot{\mathbf{q}}(t_f) = \mathbf{P}^{m_{sp}-1} = \dot{\mathbf{q}}_{t_f} \end{aligned} \quad (18)$$

**Hybrid Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{\mathbf{P}\}} J \\ \text{s. t. } & {}^a\mathbf{v}(\mathbf{P}') = (\mathbf{H}({}^a\mathbf{q}(\mathbf{P}), \boldsymbol{\theta}))^{-1} {}^a\dot{\mathbf{q}}(\mathbf{P}') \\ & {}^a\dot{\mathbf{v}}(\mathbf{P}'') = (\mathbf{H}({}^a\mathbf{q}(\mathbf{P}), \boldsymbol{\theta}))^{-1} \left( {}^a\ddot{\mathbf{q}}(\mathbf{P}'') - {}^a\mathbf{v}(\mathbf{P}') \left( \frac{\partial \mathbf{H}}{\partial t} + \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial t} \right) \right) \\ & \begin{pmatrix} {}^u\dot{\mathbf{v}} \\ {}^a\boldsymbol{\tau} \end{pmatrix} = \mathcal{G}({}^a\mathbf{q}(\mathbf{P}), {}^a\mathbf{v}(\mathbf{P}'), {}^a\dot{\mathbf{v}}(\mathbf{P}''), {}^u\boldsymbol{\tau}(\mathbf{P}), \boldsymbol{\theta}) \\ & {}^u\dot{\mathbf{q}} = {}^u\mathbf{H}({}^u\mathbf{q}, \boldsymbol{\theta}) {}^u\mathbf{v} \\ & \mathbf{y} = \mathcal{O}(\mathbf{q}(\mathbf{P}), \dot{\mathbf{q}}(\mathbf{P}'), \boldsymbol{\theta}) \\ & \mathcal{C}(\mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{0} \\ & {}^a\mathbf{q}(0) = {}^a\mathbf{P}^0 = {}^a\mathbf{q}_0 \\ & {}^a\dot{\mathbf{q}}(0) = {}^a\mathbf{P}'^0 = {}^a\dot{\mathbf{q}}_0 \\ & {}^a\mathbf{q}(t_f) = {}^a\mathbf{P}^{n_{sp}} = {}^a\mathbf{q}_{t_f} \\ & {}^a\dot{\mathbf{q}}(t_f) = {}^a\mathbf{P}^{n_{sp}-1} = {}^a\dot{\mathbf{q}}_{t_f} \\ & {}^u\mathbf{q}(0) = {}^u\mathbf{q}_0 \\ & {}^u\dot{\mathbf{q}}(0) = {}^u\dot{\mathbf{q}}_0 \\ & {}^u\mathbf{q}(t_f) = {}^u\mathbf{q}_{t_f} \\ & {}^u\dot{\mathbf{q}}(t_f) = {}^u\dot{\mathbf{q}}_{t_f} \end{aligned} \quad (19)$$

Equations (17)–(19) seeks to find the control points  $\mathbf{P}$  that minimize some prescribed objective function,  $J$ , while being subject to the dynamic constraints defined in one of (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,  $\mathcal{C}(\mathbf{y}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{0}$ . In the *hybrid dynamics* NLP formulation, equation (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  $\mathbf{P}$  and all unactuated ICs and TCs are freely defined.

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

$$J_{D1} = t_f \quad (20)$$

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

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

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

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

The solutions to (17)–(19) produces optimal motion plans under the assumption that all system properties are known (i.e. (13)–(16) are completely deterministic). The primary contribution of this work is the presentation of variants of (17)–(19) that allows (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) which 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 [45], is an efficient method for analyzing the effects of uncertainties in second order random processes [46]. This is accomplished by approximating a source of uncertainty,  $\theta$ , with an infinite series of weighted orthogonal polynomial bases called Polynomial Chaos. 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,

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

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  $\theta$  that maps the random event  $\omega \in \Omega$ , from the sample space,  $\Omega$ , to the domain of the orthogonal polynomial basis (e.g.  $\xi: \Omega \rightarrow [-1,1]$ ).

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

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

where  $w(\xi)$  is the weighting function that is equal to the joint probability density function of the random variable  $\xi$ . Also,  $\langle \Psi^j, \Psi^j \rangle = 1, \forall j$  when using *normalized basis*; *standardized basis* are constant and may be computed off-line for efficiency using (25).

Generalized Polynomial Chaos can be applied to multibody dynamical systems described by differential equations [50, 58]. 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 (24),

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

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

where  $\dot{v}_i^j(t) \in \mathbb{R}^{n_b}$  represent the gPC expansion coefficients for the  $i^{th}$  state;  $\tau_i^j(t) \in \mathbb{R}^{n_b}$  represent the gPC expansion coefficients for the  $i^{th}$  input;  $n_b \in \mathbb{N}$  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,  $\xi$ . Also, any *unknown* itemized in Table 1 has a corresponding approximation as found in (26)–(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,  $\xi$  becomes a vector of random variables,  $\xi = \{\xi_1, \dots, \xi_{n_p}\} \in \mathbb{R}^{n_p}$ , and maps the sample space,  $\Omega$ , to an  $n_p$  dimensional cuboid,  $\xi: \Omega \rightarrow [-1,1]^{n_p}$  (as in the example of Jacobi chaoses).

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

$$\Psi^j = \psi_1^{i_1} \psi_2^{i_2} \dots \psi_{n_p}^{i_{n_p}}, \quad i_k = 0 \dots p_o, k = 1 \dots n_p \quad (28)$$

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,

$$\begin{aligned} i_1 &= 0 \dots p_o, \\ i_2 &= 0 \dots (p_o - i_1), \dots, \\ i_{n_p} &= 0 \dots (p_o - i_1 - i_2 - \dots - i_{(n_p-1)}) \end{aligned}$$

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 [50, 58] but can handle large nonlinear uncertainty magnitudes.

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

**Uncertain Forward Dynamics (UFD):**

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

**Uncertain Inverse Dynamics (UID):**

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

**Uncertain Hybrid Dynamics (UHD):**

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

where the unknowns are now the unknown gPC expansion coefficients.

The Galerkin Projection Method (GPM) is a commonly used method for solving (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. [50, 51] showed that the collocation method solves formulations such as (29)–(31) by solving (13)–(16) at a set of points,  ${}_k\boldsymbol{\mu} \in \mathbb{R}^{n_p}$ ,  $k = 1 \dots n_{cp}$ , 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,

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

$${}_k\mathbf{T}_l(t; {}_k\boldsymbol{\mu}) = \sum_{j=0}^{n_b} \boldsymbol{\tau}_l^j(t)\Psi^j({}_k\boldsymbol{\mu}) \quad (33)$$

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

**Forward Dynamics Collocation Sampling:**

$${}_k\dot{\mathbf{Q}}_i(t; {}_k\boldsymbol{\mu}) = \mathcal{F}\left({}_k\mathbf{Q}_i(t; {}_k\boldsymbol{\mu}), {}_k\boldsymbol{\Theta}_r(t; {}_k\boldsymbol{\mu})\right), \quad i = 1 \dots n_s, k = 0 \dots n_{cp}, r = 1 \dots n_p \quad (34)$$

**Inverse Dynamics Collocation Sampling:**

$${}_k\mathbf{T}_i(t; {}_k\boldsymbol{\mu}) = \mathcal{F}\left(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}, {}_k\boldsymbol{\Theta}_r(t; {}_k\boldsymbol{\mu})\right), \quad i = 1 \dots n_i, k = 0 \dots n_{cp}, r = 1 \dots n_p \quad (35)$$

**Hybrid Dynamics Collocation Sampling:**

$$\begin{pmatrix} {}_k\dot{\mathbf{V}}_i(t; {}_k\boldsymbol{\mu}) \\ {}_k\mathbf{T}_l(t; {}_k\boldsymbol{\mu}) \end{pmatrix} = \mathcal{G}\left({}^a\mathbf{q}, {}^a\mathbf{v}, {}^a\dot{\mathbf{v}}, {}^u\boldsymbol{\tau}, {}_k\boldsymbol{\Theta}_r(t; {}_k\boldsymbol{\mu})\right), i = 1 \dots n_s, l = 1 \dots n_i, k = 0 \dots n_{cp}, r = 1 \dots n_p \quad (36)$$

where,

$${}_k\boldsymbol{\Theta}_r(t; {}_k\boldsymbol{\mu}) = \sum_{j=0}^{p_o} \boldsymbol{\theta}_r^j(t)\psi^j({}_k\boldsymbol{\mu}). \quad (37)$$

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 (32)–(33). In matrix notation (32)–(33) can be expressed for all states,

$$\dot{\mathbf{V}}_i = (\dot{\mathbf{v}}_i(t))^T \boldsymbol{\Psi}(\boldsymbol{\mu}), \quad i = 1 \dots n_s \quad (38)$$

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

where the matrix,

$$A_{k,j} = \psi^j({}_{k\mu}), \quad j = 0 \dots n_b, k = 0 \dots n_{cp} \quad (40)$$

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

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

$$\tau_l(t) = A^\# T_l, \quad l = 1 \dots n_i \quad (42)$$

where  $A^\#$  is the pseudo inverse of  $A$  if  $n_b < n_{cp}$ . If  $n_b = n_{cp}$ , then (41)–(42) are simply a linear solve. However, [51, 59–62] presented the least-squares collocation method (LSCM) where the stochastic state coefficients are solved for, in a least squares sense, using (41)–(42) when  $n_b < n_{cp}$ . Reference [51] also showed that as  $n_{cp} \rightarrow \infty$  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  $p_o \rightarrow \infty$ .

The nonintrusive nature of the LSCM sampling approach is arguably its greatest benefit; (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 [47–51] for more information.

## 5 MOTION PLANNING OF UNCERTAIN DYNAMICAL SYSTEMS

The deterministic motion planning formulations itemized in equations (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 (17)–(19) in Section 3, directly treats system uncertainties during the motion planning process. The formulations based on *forward*, *inverse*, and *hybrid dynamics* are,

**Forward Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{P\}} J \\ \text{s. t.} \quad & \mathcal{F}(q(\xi), v(\xi), \dot{v}(\xi), \theta(\xi)) = \tau(P) \\ & \dot{q}(\xi) = H(q(\xi), \theta(\xi))v(\xi) \\ & y(\xi) = \mathcal{O}(q(\xi), \dot{q}(\xi), \theta(\xi)) \\ & \mathcal{C}(y(\xi), \theta(\xi), \tau(P)) \leq 0 \\ & q(0; \xi) = q_0 \\ & \dot{q}(0; \xi) = \dot{q}_0 \\ & q(t_f; \xi) = q_{t_f} \\ & \dot{q}(t_f; \xi) = \dot{q}_{t_f} \end{aligned} \quad (43)$$

**Inverse Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{P\}} J \\ \text{s. t.} \quad & v(P') = (H(q(P), \theta))^{-1} \dot{q}(P') \\ & \dot{v}(P'') = (H(q(P), \theta))^{-1} \left( \ddot{q}(P'') - v(P') \left( \frac{\partial H}{\partial t} + \frac{\partial H}{\partial q} \frac{\partial q}{\partial t} + \frac{\partial H}{\partial \theta} \frac{\partial \theta}{\partial t} \right) \right) \\ & \tau(\xi) = \mathcal{F}(q(P), v(P'), \dot{v}(P''), \theta(\xi)) \\ & y(\xi) = \mathcal{O}(q(P), \dot{q}(P'), \theta(\xi)) \\ & \mathcal{C}(y(\xi), \theta(\xi), \tau(\xi)) \leq 0 \\ & q(0) = P^0 = q_0 \\ & \dot{q}(0) = P'^0 = \dot{q}_0 \\ & q(t_f) = P^{n_{sp}} = q_{t_f} \\ & \dot{q}(t_f) = P'^{n_{sp}-1} = \dot{q}_{t_f} \end{aligned} \quad (44)$$

**Hybrid Dynamics NLP Formulation:**

$$\begin{aligned} & \min_{x=\{P\}} J \\ \text{s. t.} \quad & {}^a v(P') = (H({}^a q(P), \theta))^{-1} {}^a \dot{q}(P') \\ & {}^a \dot{v}(P'') = (H({}^a q(P), \theta))^{-1} \left( {}^a \ddot{q}(P'') - {}^a v(P') \left( \frac{\partial H}{\partial t} + \frac{\partial H}{\partial q} \frac{\partial q}{\partial t} + \frac{\partial H}{\partial \theta} \frac{\partial \theta}{\partial t} \right) \right) \\ & \begin{pmatrix} {}^u \dot{v}(\xi) \\ {}^a \tau(\xi) \end{pmatrix} = \mathcal{G}({}^a q(P), {}^a v(P'), {}^a \dot{v}(P''), {}^u \tau(P), \theta(\xi)) \\ & {}^u \dot{q}(\xi) = {}^u H({}^u q(\xi), \theta(\xi)) {}^u v(\xi) \\ & y(\xi) = \mathcal{O}(q(P; \xi), \dot{q}(P'; \xi), \theta(\xi)) \end{aligned} \quad (45)$$

$$\begin{aligned}
\mathcal{C}(\mathbf{y}(\xi), \boldsymbol{\theta}(\xi), \boldsymbol{\tau}(\xi)) &\leq \mathbf{0} \\
{}^a\mathbf{q}(0) &= {}^a\mathbf{P}^0 = {}^a\mathbf{q}_0 \\
{}^a\dot{\mathbf{q}}(0) &= {}^a\mathbf{P}^0 = {}^a\dot{\mathbf{q}}_0 \\
{}^a\mathbf{q}(t_f) &= {}^a\mathbf{P}^{n_{sp}} = {}^a\mathbf{q}_{t_f} \\
{}^a\dot{\mathbf{q}}(t_f) &= {}^a\mathbf{P}^{n_{sp}-1} = {}^a\dot{\mathbf{q}}_{t_f} \\
{}^u\mathbf{q}(0; \xi) &= {}^u\mathbf{q}_0(\xi) \\
{}^u\dot{\mathbf{q}}(0; \xi) &= {}^u\dot{\mathbf{q}}_0(\xi) \\
{}^u\mathbf{q}(t_f; \xi) &= {}^u\mathbf{q}_{t_f}(\xi) \\
{}^u\dot{\mathbf{q}}(t_f; \xi) &= {}^u\dot{\mathbf{q}}_{t_f}(\xi)
\end{aligned}$$

Equations (43)–(45) are reformulations of (17)–(19) using the uncertain dynamics defined in (34)–(36). The *known* quantities of Table 1 remain deterministic, however, due to the present of the system uncertainties, all *unknown* quantities become uncertain and are modeled using the gPC techniques reviewed in Section 4. To help clarify this point, Table 1 is reproduced in Table 2 with the uncertainties explicitly illustrated.

**Table 2—Deterministic Knowns vs Uncertain Unknowns**

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

The most interesting part of (43)–(45) 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, (21)–(23) may be redefined statistically:

$$J_{S1} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} E [z_i(\tau_i(\xi, t))^2] dt = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} \sum_{j=0}^{n_b} z_i(\tau_i^j(t))^2 \langle \Psi^j, \Psi^j \rangle dt \quad (46)$$

$$J_{S2} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} E [ |z_i \tau_i(\xi, t) y_i(\xi, t)| ] dt = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} \sum_{j=0}^{n_b} |z_i \tau_i^j(t) y_i^j(t) \langle \Psi^j, \Psi^j \rangle| dt \quad (47)$$

$$J_{S3} = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} E [z_i(\dot{\tau}_i(\xi, t))^2] dt = \sum_{i=1}^{n_i} \int_{t_0=0}^{t_f} \sum_{j=0}^{n_b} z_i(\dot{\tau}_i^j(t))^2 \langle \Psi^j, \Psi^j \rangle dt \quad (48)$$

where  $\mathbf{z}$  is a vector of (optional) scalarization weights. The function (46) encapsulates the expected effort, (47) the expected power, and (48) 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, (43)–(44).

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,

$$J_{S4} = \left\| \mu_{e(t_f)} \right\| = \left\| E[\mathbf{e}(t_f; \xi)] \right\| = \left\| \mathbf{y}_{ref}(t_f) - \mathbf{y}^0(t_f) \langle \Psi^0, \Psi^0 \rangle \right\| \quad (49)$$

$$J_{S5} = \left\| \sigma_{e(t_f)}^2 \right\| = \left\| E \left[ \left( \mathbf{e}(t_f; \xi) - \mu_{e(t_f)} \right)^2 \right] \right\| = \left\| \sum_{j=0}^{n_b} (\mathbf{y}^j(t_f))^2 \langle \Psi^j, \Psi^j \rangle \right\| \quad (50)$$

where  $\mathbf{e}(t_f; \xi) = \mathbf{y}_{ref}(t_f) - \mathbf{y}(t_f; \xi)$ ; (49) is the expected value of the TC's error; (50) is the corresponding variance of the TC's error.

Due to the orthogonality of the polynomial basis, equations (46)–(50) result in a reduced set of efficient operations on their respective gPC expansion coefficients.

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

$$\mathcal{C}(t; \xi) = \underline{\mathbf{y}} \leq E[\mathbf{y}(\xi)] \leq \bar{\mathbf{y}} \quad (51)$$

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

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

$$\underline{\mathbf{y}} \leq \inf(\mathbf{y}(t; \xi)), \quad \sup(\mathbf{y}(t; \xi)) \leq \bar{\mathbf{y}} \quad (52)$$

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 standard deviation sense; collision constraints would then take the form,

$$\begin{aligned}\mu_y + \sigma_y &\leq \bar{y} \\ \underline{y} &\leq \mu_y - \sigma_y\end{aligned}\quad (53)$$

where  $std[\mathbf{y}(\xi)] = \sigma_y = \sqrt{\sum_{j=1}^{n_b} \mathbf{y}^j \langle \Psi^j, \Psi^j \rangle}$ .

Therefore, the application of the appropriate equations from (43)–(53) enables a designer to treat all possible realizations of a given uncertain system when planning motion of fully-actuated and under-actuated systems.

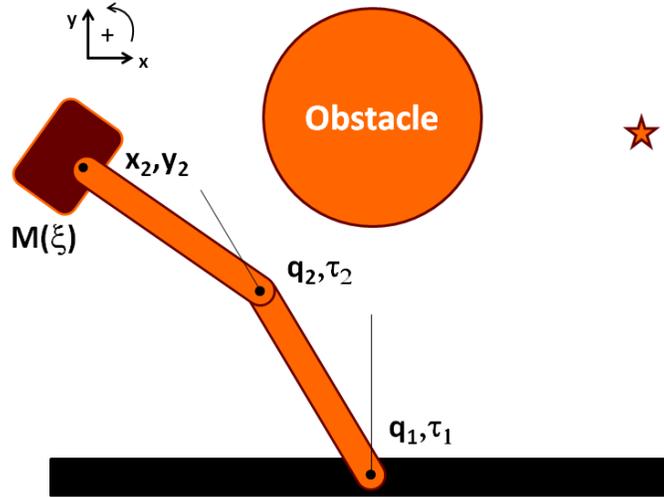
## 6 ILLUSTRATING CASE-STUDIES

This section presents case-studies which illustrating the benefits of the new motion planning framework for uncertain fully-actuated and under-actuated systems. 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. Three case-studies are presented; the first two are based on a fully-actuated serial manipulator ‘pick-and-place’ application (shown in Figure 4); the first of these uses the *forward dynamics* formulation (43); the second uses the *inverse dynamics* formulation (44). The third case-study illustrates the *hybrid dynamics* formulation (45) through an under-actuated inverting double pendulum problem (shown in Figure 11).

### 6.1 FORWARD DYNAMICS BASED UNCERTAIN MOTION PLANNING

As an illustration of (43), the serial manipulator ‘pick-and-place’ problem will be used (see Figure 4). The design objective is to minimize the effort it takes to move the manipulator from its initial configuration,  $\mathbf{q}_0$ , to the target configuration,  $\mathbf{q}_{t_f}$  in a prescribed amount of time,  $t_f$ . This results in a deterministic objective function of,  $J = \sum_{i=1}^{n_i} z_i \tau^2$ , 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  $\mathbf{q} = \{q_1, q_2\}$  are actuated with the input wrenches  $\boldsymbol{\tau} = \{\tau_1, \tau_2\}$ , the motion planning problem may be appropriately defined by (43).

By parameterizing the input wrench profiles with B-Splines, in a similar fashion as (4), (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  $\mathbf{x} = \{\mathbf{P}\}$ .



**Figure 4—A simple illustration of an uncertain fully-actuated motion planning problem; the *forward dynamics* based formulation aims to determine an *effort optimal* motion plan; 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.**

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

$$\mathcal{C}: \begin{cases} \underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}} \\ \mu_y \pm \sigma_y \leq 0 \\ -\mathcal{D}_{i,j}(\mu_y \pm \sigma_y) \leq 0 \end{cases} \quad (54)$$

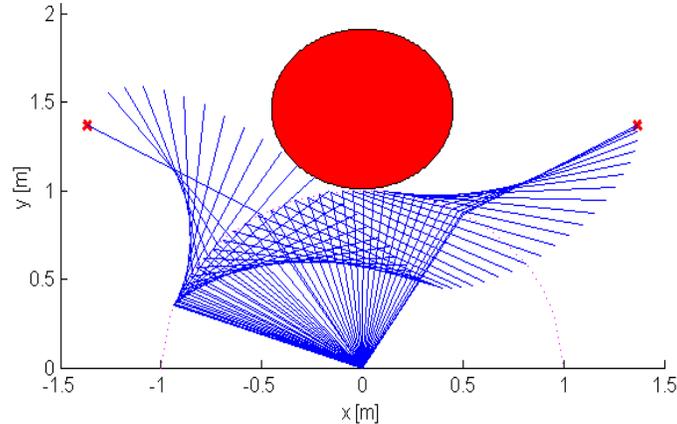
where  $i = 1,2$  and  $j = \text{obstacle}$  for the signed distance,  $\mathcal{D}_{i,j}(\mu_y \pm \sigma_y)$ , measured from each link of the serial manipulator to the obstacle calculated using the statistical mean and standard deviations of the configuration/outputs; and  $\{\underline{\boldsymbol{\tau}}, \bar{\boldsymbol{\tau}}\}$  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?*

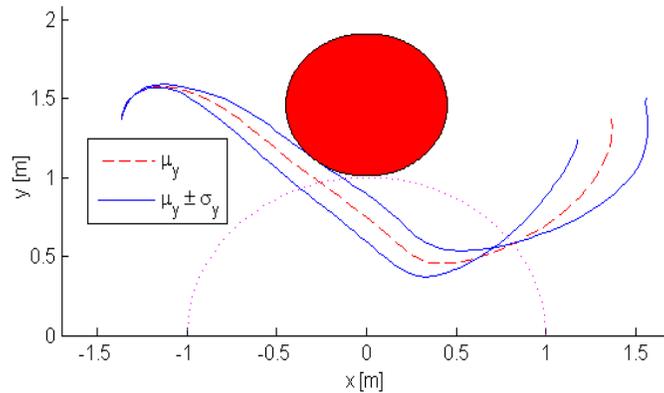
Without accounting for the uncertainty directly in the dynamics and motion planning formulations, design engineers would have a difficult time answering this question. As a result, manufacturing lines, or other applicable applications, would result in reduced yield rates potentially affecting the company’s financial *bottom-line*.

The solution to this problem with the deterministic formulation, as defined in (17), results in an *effort optimal* solution of  $J = 2770 \text{ (Nm)}^2$ ; where  $t_f = 1.5$  seconds; all system parameters are set equal to one,  $\theta_i = 1$  (with SI units); initial conditions  $\mathbf{q}(0) = \{\frac{\pi}{6}, \frac{\pi}{6}\}$  and  $\dot{\mathbf{q}}(0) = \{0, 0\}$  radians; terminal conditions  $\mathbf{q}(t_f) = \{-\frac{\pi}{6}, -\frac{\pi}{6}\}$  and  $\dot{\mathbf{q}}(t_f) = \{0, 0\}$  radians; and  $\underline{\tau} = -10, \bar{\tau} = 10 \text{ (Nm)}$ . The resulting optimal configuration time history is shown in Figure 5.



**Figure 5—The *effort optimal* configuration time histories for the deterministic serial manipulator ‘pick-and-place’ problem. This optimal solution resulted in a  $J = 2770 \text{ (Nm)}^2$  design.**

The solution from the new formulation, as defined in (43) with constraints defined by (54), results in an *effort optimal* solution of  $J = 3530 \text{ (Nm)}^2$ ; 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 pay-load 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 Figure 6; where the mean and bounding  $\mu_y \pm \sigma_y$  time histories are displayed.



**Figure 6—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  $\mu_y \pm \sigma_y$  time histories are displayed. This optimal solution resulted in a  $J = 3530 \text{ (Nm)}^2$  design.**

Therefore, the *effort optimal* solution from the uncertain problem resulted in a more conservative answer— $3530 \text{ (Nm)}^2$  as compared to  $2770 \text{ (Nm)}^2$ . This is a sensible solution; close inspection of Figure 5 shows the deterministic solution drove the configuration as close to the obstacle as possible. The introduction of uncertainty in the pay-load 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, Figure 6 shows the distribution of end-effector Cartesian position trajectory induced by the uncertain pay-load. The uncertain optimal motion plan from (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 the probability space of the uncertain mass are now guaranteed to satisfy the constraints. In other words, the *effort optimal* solution to (43) produces the minimum effort design for

the entire family of systems. Relying only on the contemporary deterministic problem formulation in (17) results in an unrealizable trajectory for a subset of the realizable systems.

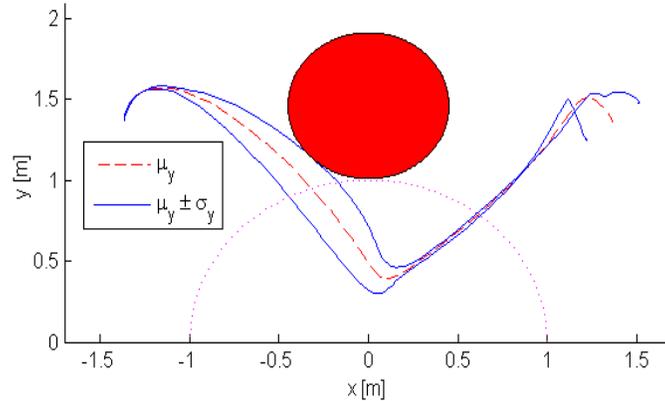
A third study provides some additional insight to what the new framework can provide. By redefining the objective function for (43) as (50) the uncertain design is no longer an *effort optimal* but *terminal variance optimal* design. In other words, the new design question is,

*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); where the standard deviation is the square root of the variance. Redesigning the motion plan using an objective function defined by (50) results in a TC error standard deviation of  $\sigma_{e(t_f)} = [0.144, 0.114]$  (m), as shown in Figure 7. Therefore, a modest reduction in the TC error standard deviation was realized, however, the effort of the new design increased from 3530 (Nm)<sup>2</sup> to 5910 (Nm)<sup>2</sup>. 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 (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 (44); this is illustrated in the next section.



**Figure 7—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  $\mu_y \pm \sigma_y$  time histories are displayed. This optimal solution resulted in a  $J = 5910$  (Nm)<sup>2</sup> design.**

## 6.2 INVERSE DYANAMICS BASED UNCERTAIN MOTION PLANNING

As an illustration of (44), the serial manipulator “pick-and-place” problem is re-used (see Figure 4). The design objective is to minimize the time it takes to move the manipulator from its initial configuration,  $\mathbf{q}_0$ , to the target configuration,  $\mathbf{q}_{t_f}$ . 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  $\mathbf{q} = \{q_1, q_2\}$  are actuated with the input wrenches  $\boldsymbol{\tau} = \{\tau_1, \tau_2\}$ , the motion planning problem may be appropriately defined by (44).

By parameterizing the deterministic joint trajectories with B-Splines, as in (4), (44) 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  $\mathbf{x} = \{\mathbf{P}, t_f\}$ .

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

$$\mathbf{c}: \begin{cases} \mu_{\tau_i} + \sigma_{\tau_i} \leq \bar{\tau} \\ \underline{\tau} \leq \mu_{\tau_i} - \sigma_{\tau_i} \\ -y_1 \leq 0 \\ -y_2 \leq 0 \\ -\mathcal{D}_{i,j} \leq 0 \end{cases} \quad (55)$$

where  $i = 1, 2$  and  $j = \text{obstacle}$  for the signed distance,  $\mathcal{D}_{i,j}$ , 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 standard deviations, as in (53), 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 standard deviation, as in (55), 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.

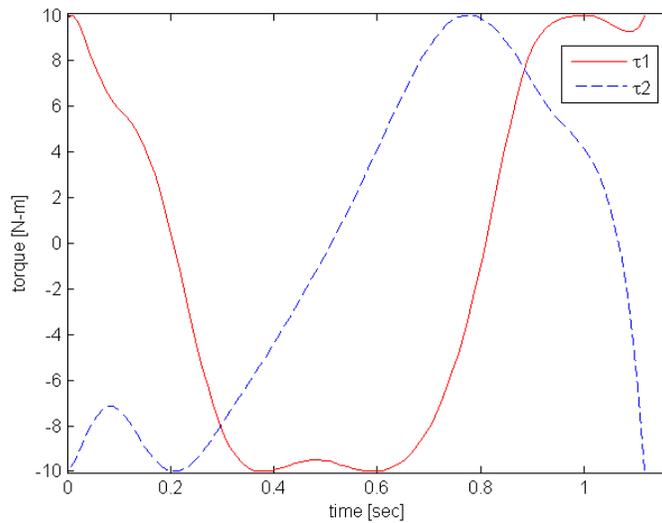
This formulation allows a design engineer to answer the question,

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

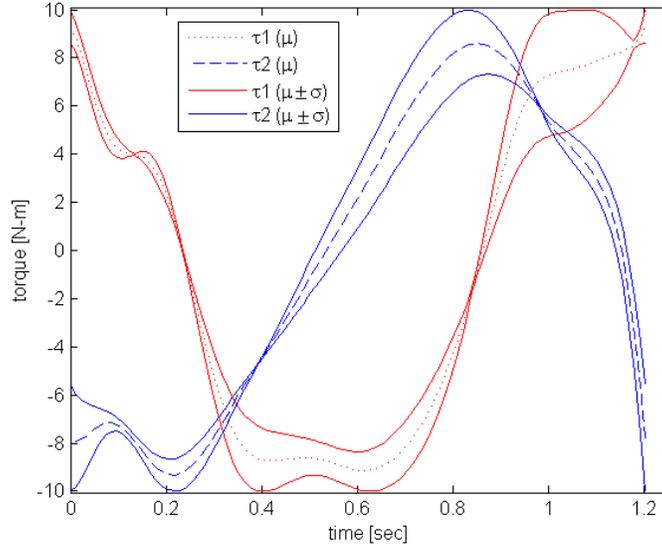
Without accounting for the uncertainty directly in the dynamics and motion planning formulations, design engineers would have a difficult time answering this question. As a result, manufacturing lines, or other applicable applications, would result in reduced yield rates potentially affecting the company’s financial *bottom-line*.

The solution to this problem with the deterministic formulation, as defined in (18), results in a *time optimal* solution of  $t_f = 1.12$  seconds; where all system parameters are set equal to one,  $\theta_i = 1$  (with SI units); with initial conditions  $\mathbf{q}(0) = \{\frac{\pi}{6}, \frac{\pi}{6}\}$  and  $\dot{\mathbf{q}}(0) = \{0, 0\}$  radians; terminal conditions  $\mathbf{q}(t_f) = \{-\frac{\pi}{6}, -\frac{\pi}{6}\}$  and  $\dot{\mathbf{q}}(t_f) = \{0, 0\}$  radians; and  $\underline{\tau} = -10, \bar{\tau} = 10$  (Nm). The resulting optimal input wrench time history is shown in Figure 8.

The solution from the new formulation, as defined in (44) with constraints defined by (55), results in a *time optimal* solution of  $t_f = 1.2$  seconds; 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 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 Figure 9; where each input wrench is displaying its mean value and bounding  $\mu_{\tau_i} \pm \sigma_{\tau_i}$  time histories. Also, the resulting configuration time history for the optimal uncertain motion plan is shown in Figure 10.

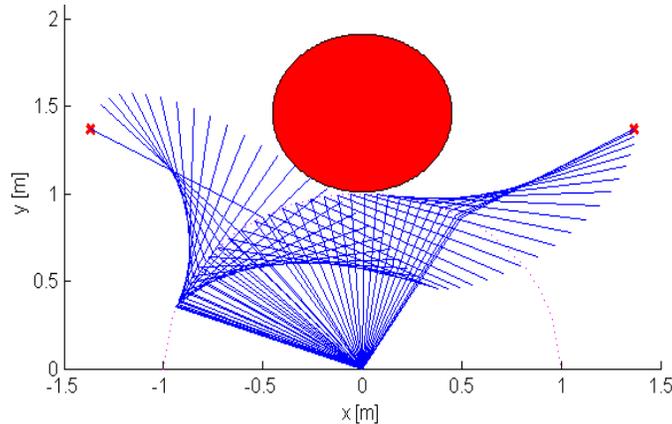


**Figure 8—The *time optimal* input wrench time histories for the deterministic serial manipulator ‘pick-and-place’ problem based on the *uncertain inverse dynamics* NLP. This optimal solution resulted in a  $t_f = 1.12$  (s).**



**Figure 9**—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  $\mu_{\tau_i} \pm \sigma_{\tau_i}$  time histories. 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 seconds as compared to 1.12 seconds). This is a sensible solution; close inspection of Figure 8 shows the deterministic solution drove the input wrenches to their extreme bounds of  $\pm 10$  (Nm) 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, Figure 9 shows the distribution of input wrenches induced by the uncertain mass. The uncertain optimal motion plan from (44) 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 the probability space of the uncertain mass are now guaranteed to satisfy the constraints. In other words, the *time optimal* solution to (44) produces the minimum time for the entire family of systems. Relying only on the contemporary deterministic problem formulation in (18) results in an unrealizable trajectory for a subset of the realizable systems.



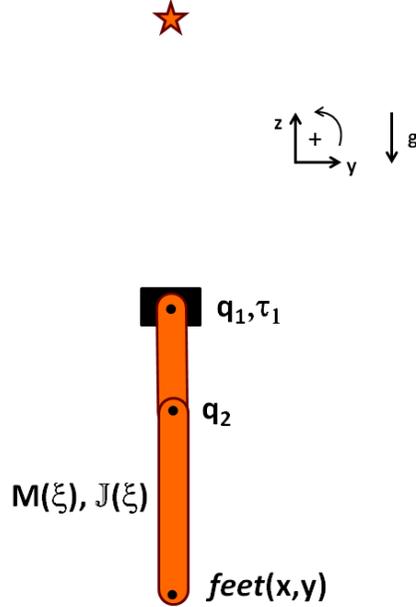
**Figure 10**—The final optimal configuration time history of the uncertain serial manipulator ‘pick-and-place’ application involving collision avoidance and actuator constraints design with the *uncertain inverse dynamics* NLP.

A final observation is that the *uncertain inverse dynamics* motion planning framework embodied in (44) is most applicable to configuration/position controlled systems, where states are prescribed as they are in (4). However, force controlled systems may be better designed through application of (43) based on *uncertain forward dynamics* as illustrated in the previous section, Section 6.1.

### 6.3 HYBRID DYNAMICS

As an illustration of (45), an inverting double pendulum problem will be used (see Figure 11). The design objective is to minimize the power it takes to move the manipulator from its initial hanging configuration,  $\mathbf{q}_0$ , to the target inverted

configuration,  $\mathbf{q}_{t_f}$ . The double pendulum is an under-actuated system, where only joint  $q_1$  is actuated (by input wrench  $\tau_1$ ), and the mass of the second link is uncertain, therefore, the motion planning problem may be appropriately defined by (45).



**Figure 11—A simple illustration of the under-actuated *uncertain hybrid dynamics* motion planning formulation; this problem aims to determine a *power optimal* motion plan subject to input wrench and terminal condition constraints. This is an uncertain system due to the uncertain mass of the payload.**

By parameterizing the actuated state profiles with B-Splines, as in (4), and using the *hybrid dynamics* defined in (16), (45) 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  $\mathbf{x} = \{\mathbf{P}, t_f\}$ . Assuming a *soft* terminal error expected value condition is used, the objective function becomes  $J = a \cdot J_{S2} + b \cdot J_{S4}$  from (47)–(49); 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 (50) that must be satisfied. Implementing both of these design constraints as *hard* constraints takes the form,

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

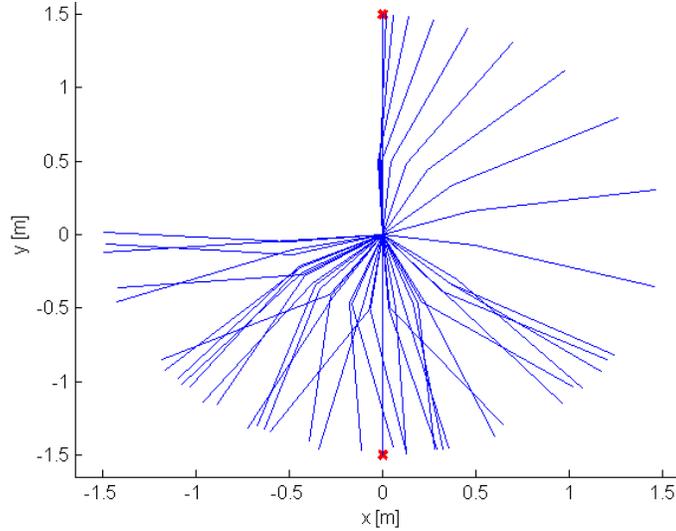
where  $\{\underline{\tau}, \bar{\tau}\}$  are the minimum/maximum input bounds respectively;  $\bar{\sigma}_{e(t_f)}^2$  is the maximum terminal error variance.

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. As a result, manufacturing lines, or other applicable applications, would result in reduced yield rates potentially affecting a company's financial *bottom-line*.

The solution to this problem with the deterministic formulation, as defined in (17), results in an *power optimal* solution of  $J_{S1} = 1060$  ( $W$ ) with  $t_f = 5.66$  seconds; all system parameters are set equal to one,  $\theta_i = 1$  (with SI units) except the length of the first link is set to  $0.5$  ( $m$ ); initial conditions  $\mathbf{q}(0) = \{-\pi, 0\}$  and  $\dot{\mathbf{q}}(0) = \{0, 0\}$  radians; terminal conditions  $\mathbf{q}(t_f) = \{0, 0\}$  and  $\dot{\mathbf{q}}(t_f) = \{0, 0\}$  radians; and the input limits are  $\underline{\tau} = -10$ ,  $\bar{\tau} = 10$  ( $N \cdot m$ ). The resulting optimal motion plan's configuration time history is shown in Figure 12.

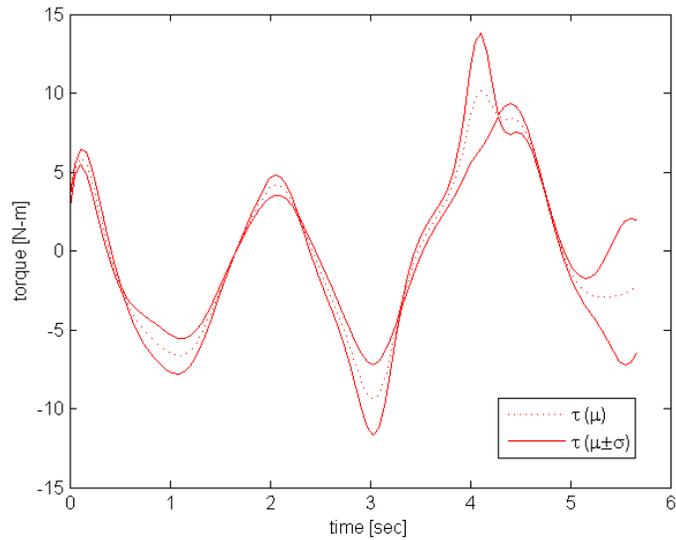


**Figure 12—The *power optimal* configuration time history for the deterministic inverting double pendulum. This optimal solution resulted in a 1060 (W) design.**

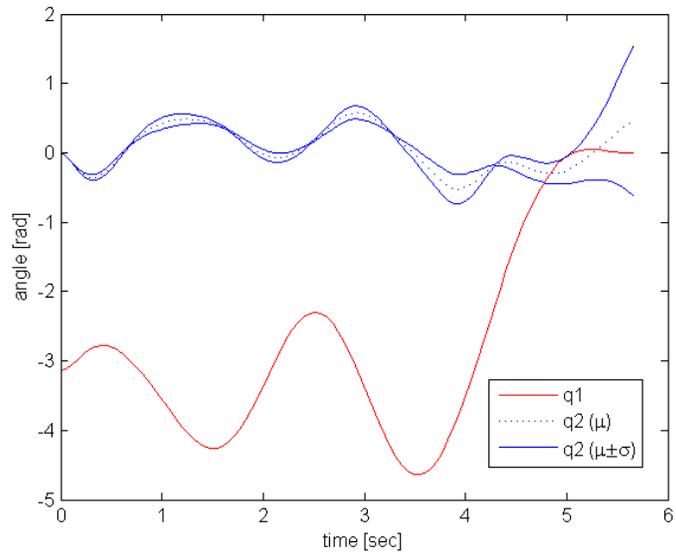
The value of the new framework is best illustrated by applying the deterministically designed motion profile to an uncertain system. Figure 13 and Figure 14 show the results of the deterministic motion plan applied to a system with a single uncertainty; the second link has an uncertain mass with  $\mu_{m_2} = 1$  (kg) and  $\sigma_{m_2}^2 = 0.5$  (kg<sup>2</sup>). Figure 13 shows that the resulting input wrench profile exceeds both the upper and lower bounding constraints of  $\underline{\tau} = -10, \bar{\tau} = 10$  (N · m). Additionally, Figure 14 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 (45) with constraints defined by (56); where  $\bar{\sigma}_{e(t_f)}^2 = 0.01$  (m<sup>2</sup>). This results in a *power optimal* solution of  $J_{S2} = 310$  (W) with  $t_f = 4.46$  seconds; where the same uncertain second link mass is reused. The resulting motion plan's optimal uncertain configuration time history is illustrated in Figure 15; where the bounding  $\{\mu_y - \sigma_y$  (red),  $\mu_y + \sigma_y$  (blue) $\}$  configuration time histories are displayed. The Euclidean norm of the *soft* expected value terminal configuration constraint was very acceptable,  $\|E[e(t_f)]\| = 2.61e - 6$  (m). Figure 16 shows that the input wrench constraints for the entire probability space were satisfied in a standard deviation sense. Figure 17 show that the specified terminal error variance was also satisfied,  $\sigma_{e(t_f)}^2 = 0.00321 \leq \bar{\sigma}_{e(t_f)}^2 = 0.01$  (m<sup>2</sup>).

The reduced power of the uncertain design, as compared to the deterministic design, makes sense in that the expected input wrench values,  $E[\tau_1]$ , of the uncertain design (as shown in Figure 16), are lower than those in the deterministic design (as shown in Figure 13). 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.



**Figure 13—The uncertain input wrench time history for the deterministically design motion plan applied to an uncertain inverting double pendulum. The presence of the uncertainty results in both the maximum and minimum input limits being exceeded.**



**Figure 14—The joint time histories for the deterministically design motion plan applied to an uncertain inverting double pendulum. The presence of the uncertainty results in the expected terminal error condition not being satisfied with excessive variance.**

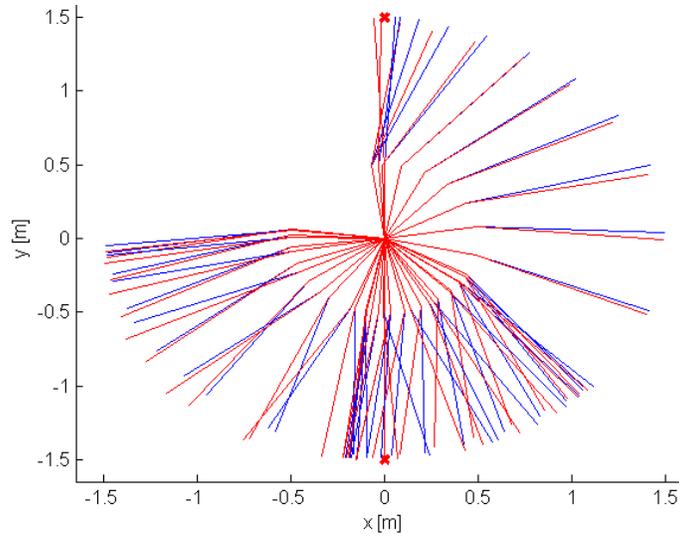


Figure 15—The *power optimal* configuration time history for the uncertain inverting double pendulum based on *uncertain hybrid dynamics* NLP. This optimal solution resulted in a **310 (W)** design.

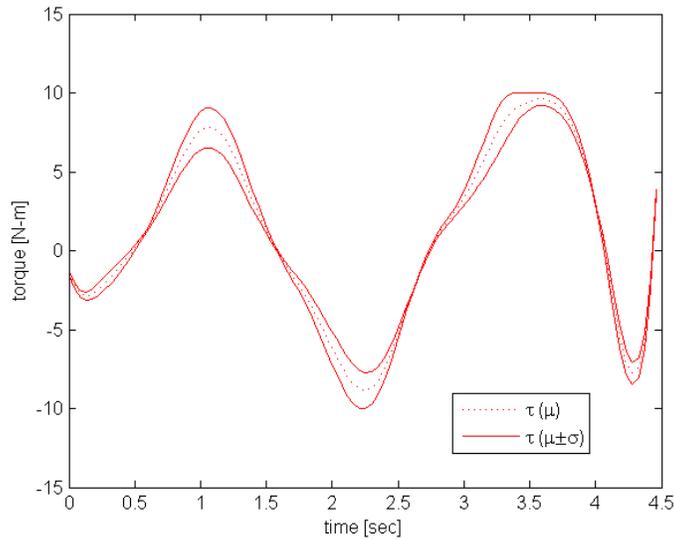
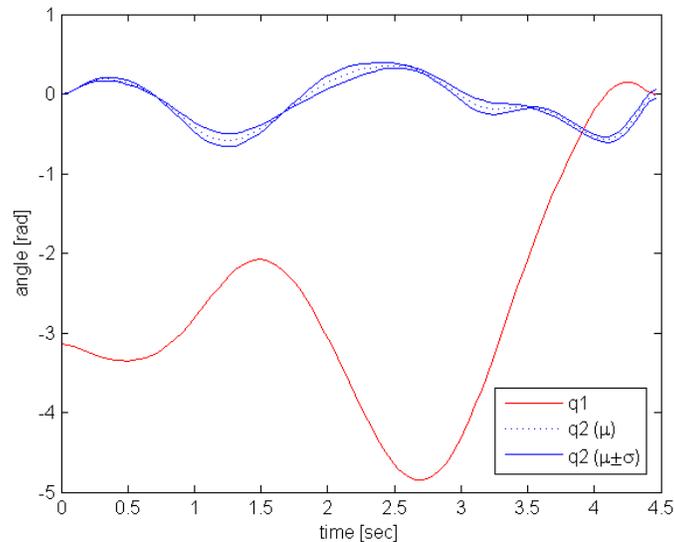


Figure 16—The uncertain input wrench time history resulting from the motion plan generated by the new *uncertain hybrid dynamics* NLP. Both the maximum and minimum input limits were satisfied, in a standard deviation sense, for all systems within the probability space.



**Figure 17— The joint time histories resulting from the motion plan generated by the new *uncertain hybrid dynamics* NLP. The resulting terminal error variance satisfies the specification;  $\sigma_{e(t_f)}^2 = 0.0032 \leq \bar{\sigma}_{e(t_f)}^2 = 0.01 (m^2)$ .**

## 7 CONCLUSIONS

This work has presented a new nonlinear programming based framework for motion planning that treats uncertain fully-actuated and under-actuated 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.

## ACKNOWLEDGEMENTS

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

## REFERENCES

- [1] Man, X., Swan, C., and Rahmatalla, S., 2006, "A Clothing Modeling Framework for Uniform and Armor System Design," eds., pp. 17–22.
- [2] 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," *Structural and Multidisciplinary Optimization*, 41(3), pp. 465-479.
- [3] 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," *Journal of biomechanical engineering*, 130(pp. 031002.
- [4] 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," *Structural and Multidisciplinary Optimization*, 42(1), pp. 1-23.
- [5] Company, F. M., May 19, 2010, Ford Recruits Virtual Soldier to Boost Quality; Santos Feels the Same Strains That Humans Do, <http://www.prnewswire.com/news-releases/ford-recruits-virtual-soldier-to-boost-quality-santos-feels-the-same-strains-that-humans-do-94221984.html>
- [6] Park, J., 2007, *Industrial Robotics, Programming, Simulation and Applications*, Verlag, Croatia, Optimal Motion Planning for Manipulator Arms Using Nonlinear Programming.
- [7] Chong Jin, O., and Gilbert, E. G., 1996, "Growth Distances: New Measures for Object Separation and Penetration," *Robotics and Automation, IEEE Transactions on*, 12(6), pp. 888-903.
- [8] Park, F., Bobrow, J., and Ploen, S., 1995, "A Lie Group Formulation of Robot Dynamics," *The International Journal of Robotics Research*, 14(6), pp. 609.
- [9] Ploen, S., 1997, "Geometric Algorithms for the Dynamics and Control of Multibody Systems," Ph.D. thesis, University Of California,

- [10] Park, F. C., and Bobrow, J. E., 1994, "A Recursive Algorithm for Robot Dynamics Using Lie Groups," eds., pp. 1535-1540 vol.2.
- [11] Martin, B., and Bobrow, J., 1999, "Minimum-Effort Motions for Open-Chain Manipulators with Task-Dependent End-Effector Constraints," *The International Journal of Robotics Research*, 18(2), pp. 213.
- [12] Sohl, G. A., and Bobrow, J. E., 2001, "A Recursive Multibody Dynamics and Sensitivity Algorithm for Branched Kinematic Chains," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, 123(Copyright 2002, IEE), pp. 391-9.
- [13] Bobrow, J. E., and Sohl, G. A., "On the Reliable Computation of Optimal Motions for Underactuated Manipulators," pp.
- [14] Bobrow, J., Martin, B., Sohl, G., Wang, E., Park, F., and Kim, J., 2001, "Optimal Robot Motions for Physical Criteria," *Journal of Robotic systems*, 18(12), pp. 785-795.
- [15] Sohl, G., 2000, "Optimal Dynamic Motion Planning for Underactuated Robots," Ph.D. thesis,
- [16] Wang, C., Timoszyk, W., and Bobrow, J., 1999, "Weightlifting Motion Planning for a Puma 762 Robot," eds., pp. 480-485.
- [17] Wang, C., 2001, "Dynamic Motion Planning for Robot Manipulators Using B-Splines," Ph.D. thesis, University Of California,
- [18] Junggon, K., Jonghyun, B., and Park, F. C., 1999, "Newton-Type Algorithms for Robot Motion Optimization," eds., 3, pp. 1842-1847 vol.3.
- [19] Bobrow, J. E., Park, F. C., and Sideris, A., 2006, *Fast Motions in Biomechanics and Robotics*, Springer Berlin / Heidelberg, Recent Advances on the Algorithmic Optimization of Robot Motion.
- [20] Lee, S. H., Kim, J., Park, F. C., Kim, M., and Bobrow, J. E., 2005, "Newton-Type Algorithms for Dynamics-Based Robot Movement Optimization," *Robotics, IEEE Transactions on*, 21(4), pp. 657-667.
- [21] Xiang, Y., Arora, J., and Abdel-Malek, K., 2009, "Optimization-Based Motion Prediction of Mechanical Systems: Sensitivity Analysis," *Structural and Multidisciplinary Optimization*, 37(6), pp. 595-608.
- [22] 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," *Optimization*, 1(pp. 2489.
- [23] 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," eds., pp. 16-18.
- [24] Xiang, Y., Arora, J. S., Rahmatalla, S., and Abdel-Malek, K., 2009, "Optimization-Based Dynamic Human Walking Prediction: One Step Formulation," *International Journal for Numerical Methods in Engineering*, 79(6), pp. 667-695.
- [25] Xiang, Y., 2008, "Optimization-Based Dynamic Human Walking Prediction," Ph.D. thesis, University of Iowa, Iowa City, IA.
- [26] Diehl, M., Ferreau, H., and Haverbeke, N., 2009, "Efficient Numerical Methods for Nonlinear Mpc and Moving Horizon Estimation," *Nonlinear Model Predictive Control*, pp. 391-417.
- [27] Biegler, L. T., 2003, "Optimization of Ode/Dae Constrained Models," Technical Report No.
- [28] Park, J., and Park, F., 2006, *Advances in Robot Kinematics*, Springer Netherlands, A Convex Optimization Algorithm for Stabilizing Whole-Body Motions of Humanoid Robots.
- [29] Suleiman, W., Yoshida, E., Laumond, J. P., and Monin, A., 2007, "On Humanoid Motion Optimization," eds., pp. 180-187.
- [30] Lim, B., Kim, B., Park, F. C., and Hong, D. W., 2009,
- [31] Joe Hays, Dennis Hong, Corina Sandu, and Sandu, A., 2010,
- [32] Lavalle, S., 2006, *Planning Algorithms*, Cambridge Univ Pr,
- [33] Choset, H., 2005, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, The MIT Press,
- [34] Karaman, S., Frazzoli, E., and Altendorfer, R., 2010, "Incremental Sampling-Based Algorithms for Optimal Motion Planning," *Arxiv preprint arXiv:1005.0416*, pp.
- [35] Barraquand, J., and Ferbach, P., 2002, "Motion Planning with Uncertainty: The Information Space Approach," eds., 2, pp. 1341-1348.
- [36] 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.
- [37] Erdmann, M., 1984, "On Motion Planning with Uncertainty," AITR-810, pp.
- [38] Kewlani, G., Ishigami, G., and Iagnemma, K., 2009, "Stochastic Mobility-Based Path Planning in Uncertain Environments," eds., pp. 1183-1189.
- [39] Hays, J., Sandu, A., and Sandu, C., 2011 (submitted), "Motion Planning of Uncertain Fully-Actuated Dynamical Systems—an Inverse Dynamics Formulation," eds., Washington, DC, USA, pp.
- [40] Hays, J., Sandu, A., and Sandu, C., 2011 (submitted), "Motion Planning of Uncertain Fully-Actuated Dynamical Systems—a Forward Dynamics Formulation," eds., Washington, DC, USA, pp.
- [41] Hays, J., Sandu, A., Sandu, C., and Hong, D., 2011 (submitted), "Motion Planning of Uncertain under-Actuated Dynamical Systems—a Hybrid Dynamics Formulation," eds., Denver, CO, USA, pp.
- [42] Papoulis, A., Pillai, S., and Unnikrishna, S., 2002, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill New York,
- [43] Rensburg, E., and Torrie, G., 1993, "Estimation of Multidimensional Integrals: Is Monte Carlo the Best Method?," *Journal of Physics A: Mathematical and General*, 26(pp. 943.
- [44] Bratley, P., Fox, B., and Niederreiter, H., 1992, "Implementation and Tests of Low-Discrepancy Sequences," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2(3), pp. 213.
- [45] Wiener, N., 1938, "The Homogeneous Chaos," *American Journal of Mathematics*, 60(4), pp. 897-936.

- [46] Xiu, D., and Karniadakis, G., 2003, "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," pp.
- [47] Xiu, D., 2009, "Fast Numerical Methods for Stochastic Computations: A Review," *Communications in Computational Physics*, 5(2-4), pp. 242-272.
- [48] Xiu, D., and Hesthaven, J. S., 2005, "High-Order Collocation Methods for Differential Equations with Random Inputs," *SIAM Journal on Scientific Computing*, 27(3), pp. 1118-1139.
- [49] Xiu, D., 2007, "Efficient Collocational Approach for Parametric Uncertainty Analysis," *Communications in Computational Physics*, 2(2), pp. 293-309.
- [50] Sandu, A., Sandu, C., and Ahmadian, M., 2006, "Modeling Multibody Systems with Uncertainties. Part I: Theoretical and Computational Aspects," *Multibody System Dynamics*, 15(4), pp. 369-391.
- [51] Cheng, H., and Sandu, A., 2009, "Efficient Uncertainty Quantification with the Polynomial Chaos Method for Stiff Systems," *Mathematics and Computers in Simulation*, 79(11), pp. 3278-3295.
- [52] Wan, X., and Karniadakis, G., 2005, "An Adaptive Multi-Element Generalized Polynomial Chaos Method for Stochastic Differential Equations," *Journal of Computational Physics*, 209(2), pp. 617-642.
- [53] Wan, X., and Karniadakis, G., "Adaptive Numerical Solutions of Stochastic Differential Equations," pp.
- [54] Wan, X., and Karniadakis, G., 2006, "Beyond Wiener-Askey Expansions: Handling Arbitrary Pdfs," *Journal of Scientific Computing*, 27(1), pp. 455-464.
- [55] Wan, X., and Karniadakis, G., 2007, "Multi-Element Generalized Polynomial Chaos for Arbitrary Probability Measures," *SIAM Journal on Scientific Computing*, 28(3), pp. 901-928.
- [56] 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.
- [57] Foo, J., and Karniadakis, G. E., 2010, "Multi-Element Probabilistic Collocation Method in High Dimensions," *Journal of Computational Physics*, 229(5), pp. 1536-1557.
- [58] Sandu, C., Sandu, A., and Ahmadian, M., 2006, "Modeling Multibody Systems with Uncertainties. Part II: Numerical Applications," *Multibody System Dynamics*, 15(3), pp. 241-262.
- [59] Cheng, H., and Sandu, A., 2007, "Numerical Study of Uncertainty Quantification Techniques for Implicit Stiff Systems," eds., Winston-Salem, NC, USA, pp. 367-372.
- [60] Cheng, H., and Sandu, A., 2009, "Uncertainty Quantification in 3d Air Quality Models Using Polynomial Chaos," *Environmental Modeling and Software*, 24(8), pp. 917-925.
- [61] Cheng, H., and Sandu, A., 2009, "Uncertainty Apportionment for Air Quality Forecast Models," eds., Honolulu, HI, USA, pp. 956-960.
- [62] Cheng, H., and Sandu, A., 2010, "Collocation Least-Squares Polynomial Chaos Method," eds., Orlando, FL, USA, pp. 80.
- [63] Blanchard, E., 2010, "Polynomial Chaos Approaches to Parameter Estimation and Control Design for Mechanical Systems with Uncertain Parameters," Ph.D. thesis,
- [64] Blanchard, E., Sandu, A., and Sandu, C., 2007, "Parameter Estimation Method Using an Extended Kalman Filter," eds., Fairbanks, Alaska, USA, pp. 23-26.
- [65] Blanchard, E., Sandu, A., and Sandu, C., 2009, "Parameter Estimation for Mechanical Systems Via an Explicit Representation of Uncertainty," *Engineering Computations*, 26(5), pp. 541-569.
- [66] Blanchard, E., Sandu, A., and Sandu, C., 2010, "Polynomial Chaos-Based Parameter Estimation Methods Applied to a Vehicle System," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 224(1), pp. 59-81.
- [67] Blanchard, E., Sandu, A., and Sandu, C., 2010, "Polynomial Chaos Based Method for the Lqr Problem with Uncertain Parameters in the Formulation," eds., Montreal, CA, pp.
- [68] Blanchard, E., Sandu, C., and Sandu, A., 2007, "A Polynomial-Chaos-Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems," eds., Las Vegas, NV, USA, pp. 4-7.
- [69] 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," eds., San Diego, CA, USA, pp.
- [70] Blanchard, E., and Sandu, D., 2007, "A Polynomial Chaos Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems-Part II: Applications to Vehicle Systems," pp.
- [71] Blanchard, E., and Sandu, D., 2007, "A Polynomial Chaos Based Bayesian Approach for Estimating Uncertain Parameters of Mechanical Systems-Part I: Theoretical Approach," pp.
- [72] Blanchard, E. D., Sandu, A., and Sandu, C., 2010, "A Polynomial Chaos-Based Kalman Filter Approach for Parameter Estimation of Mechanical Systems," *Journal of Dynamic Systems, Measurement, and Control*, 132(6), pp. 061404.
- [73] Pence, B., Hays, J., Fathy, H., Sandu, C., and Stein, J., 2011 (submitted), "Vehicle Sprung Mass Estimation for Rough Terrain," *International Journal of Vehicle Design, Special Issue on Modeling and Simulation of Ground Vehicle Systems*(pp.
- [74] 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," eds., n PART A, pp. 857-864.
- [75] Pence, B. L., Fathy, H. K., and Stein, J. L., 2010,
- [76] Pence, B. L., Fathy, H. K., and Stein, J. L., 2010,
- [77] Southward, S., 2007, "Real-Time Parameter Id Using Polynomial Chaos Expansions," eds., Seattle, WA, USA, pp.
- [78] Shimp Iii, S., 2008, "Vehicle Sprung Mass Parameter Estimation Using an Adaptive Polynomial-Chaos Method," pp.

- [79] Marzouk, Y., and Xiu, D., 2009, "A Stochastic Collocation Approach to Bayesian Inference in Inverse Problems," *Communications in Computational Physics*, 6(pp. 826–847.
- [80] Marzouk, Y. M., Najm, H. N., and Rahn, L. A., 2007, "Stochastic Spectral Methods for Efficient Bayesian Solution of Inverse Problems," *Journal of Computational Physics*, 224(2), pp. 560-586.
- [81] Price, D., 2008, "Estimation of Uncertain Vehicle Center of Gravity Using Polynomial Chaos Expansions," Ph.D. thesis, Virginia Polytechnic Institute and State University,
- [82] Smith, A., Monti, A., and Ponci, F., 2007, "Indirect Measurements Via a Polynomial Chaos Observer," *IEEE Transactions on Instrumentation and Measurement*, 56(3), pp. 743-752.
- [83] Li, J., and Xiu, D., 2009, "A Generalized Polynomial Chaos Based Ensemble Kalman Filter with High Accuracy," *Journal of Computational Physics*, 228(15), pp. 5454-5469.
- [84] 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," eds., 6, pp. 6005-13.
- [85] Templeton, B., 2009, "A Polynomial Chaos Approach to Control Design," Ph.D. thesis,
- [86] Smith, A., Monti, A., and Ponci, F., 2006, "Robust Controller Using Polynomial Chaos Theory," eds., 5, pp.
- [87] Prempraneerach, P., Hover, F., Triantafyllou, M., and Karniadakis, G., 2010, "Uncertainty Quantification in Simulations of Power Systems: Multi-Element Polynomial Chaos Methods," *Reliability Engineering & System Safety*, pp.
- [88] Kewlani, G., and Iagnemma, K., 2009, "A Multi-Element Generalized Polynomial Chaos Approach to Analysis of Mobile Robot Dynamics under Uncertainty," eds., pp. 1177-1182.
- [89] Greenwood, D., 2003, *Advanced Dynamics*, Cambridge Univ Pr,
- [90] Murray, R., Li, Z., Sastry, S., and Sastry, S., 1994, *A Mathematical Introduction to Robotic Manipulation*, CRC,
- [91] Nkravesh, P. E., 2004, *Product Engineering*, Springer, An Overview of Several Formulations for Multibody Dynamics.
- [92] Haug, E. J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems. Vol. 1: Basic Methods*, Allyn & Bacon, Inc.,
- [93] Piegl, L. A., and Tiller, W., 1997, *The Nurbs Book*, Springer Verlag,