



Cedarville University  
**DigitalCommons@Cedarville**

---

Engineering and Computer Science Faculty  
Publications

School of Engineering and Computer Science

---

8-2011

# Hybrid Gauss Pseudospectral and Generalized Polynomial Chaos Algorithm to Solve Stochastic Optimal Control Problems

G. C. Cottrill

Frederick G. Harmon  
*Cedarville University*, [fharmon@cedarville.edu](mailto:fharmon@cedarville.edu)

Follow this and additional works at: [http://digitalcommons.cedarville.edu/  
engineering\\_and\\_computer\\_science\\_publications](http://digitalcommons.cedarville.edu/engineering_and_computer_science_publications)

 Part of the [Engineering Commons](#)

---

## Recommended Citation

Cottrill, G. C. and Harmon, Frederick G., "Hybrid Gauss Pseudospectral and Generalized Polynomial Chaos Algorithm to Solve Stochastic Optimal Control Problems" (2011). *Engineering and Computer Science Faculty Publications*. 188.  
[http://digitalcommons.cedarville.edu/engineering\\_and\\_computer\\_science\\_publications/188](http://digitalcommons.cedarville.edu/engineering_and_computer_science_publications/188)

This Conference Proceeding is brought to you for free and open access by DigitalCommons@Cedarville, a service of the Centennial Library. It has been accepted for inclusion in Engineering and Computer Science Faculty Publications by an authorized administrator of DigitalCommons@Cedarville. For more information, please contact [digitalcommons@cedarville.edu](mailto:digitalcommons@cedarville.edu).



# Hybrid Gauss Pseudospectral and Generalized Polynomial Chaos Algorithm to Solve Stochastic Trajectory Optimization Problems

Maj Gerald C. Cottrill\*

*Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433, USA*

Lt Col Frederick G. Harmon<sup>†</sup>

*Air Force Institute of Technology, Wright-Patterson AFB, OH, 45433, USA*

**A** numerical algorithm combining the Gauss Pseudospectral Method (GPM) with a Generalized Polynomial Chaos (gPC) method to solve nonlinear stochastic optimal control problems with constraint uncertainties is presented. The GPM and gPC have been shown to be spectrally accurate numerical methods for solving deterministic optimal control problems and stochastic differential equations, respectively. The gPC uses collocation nodes to sample the random space, which are then inserted into the differential equations and solved using standard solvers to generate a set of deterministic solutions used to characterize the distribution of the solution by constructing a polynomial representation of the output as a function of uncertain parameters. The proposed algorithm investigates using GPM optimization software in place of deterministic differential equation solvers traditionally used in the gPC, providing minimum cost deterministic solutions that meet path, control, and boundary constraints. A trajectory optimization problem is considered where the objectives are to find the path through a two-dimensional space that minimizes the probability a vehicle will be 'killed' by lethal threats whose locations are uncertain and to characterize the effects those uncertainties have on the solution by estimating the statistical properties.

## Nomenclature

$\hat{z}$	gPC expansion coefficients
$E$	Expected value operator, expectation, expected value
$P$	Projection operator
$C$	Path constraints
$D$	Differentiation matrix
$p$	Stochastic input parameters
$U$	Polynomial approximation of the control vector
$u$	Control vector
$X$	Polynomial approximation of the state vector
$x$	State vector
$z$	Outputs/observables of gPC method
$\mathcal{A}$	Sigma-algebra of probability space
$\mathcal{P}$	Measure of probability space
$cov$	Covariance
$d$	Degree of the one-dimensional polynomial basis elements
$F$	Function of states and random inputs in gPC formulation
$f$	General function of states and controls defining the system dynamics
$g$	Function of initial conditions in gPC formulation

\*Ph.D. Student, Department of Aeronautical and Astronautical Engineering, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, Student Member.

<sup>†</sup>Assistant Professor of Aeronautical Engineering, Department of Aeronautical and Astronautical Engineering, 2950 Hobson Way, Wright-Patterson AFB, OH, 45433, Senior Member.

$g$	General functional of states and controls used in integral cost functional term
$H$	Hermite polynomial
$h$	Orthogonality condition constant
$i$	Counting index
$J$	Cost functional
$k$	Counting index
$L$	Lagrange interpolating polynomial
$M$	Total number of basis elements in the N-dimensional polynomial space
$m$	Counting index
$N$	Number of collocation points
$NM$	Nautical Miles
$P$	Total degree of the polynomials making up the N-variate polynomial space
$Q$	Number of collocation points in the N-dimensional probability space
$q$	Number of collocation points in one-dimensional probability space
$R_{min}$	Minimum turn radius, ft
$t_0$	Initial time
$t_f$	Final or terminal time
$V$	Speed, kts
$var$	Variance
$W$	One-dimensional orthogonal polynomial space
$w$	Quadrature weight
$x_1$	Position with respect to $x_1$ axis, $NM$
$x_2$	Position with respect to $x_2$ axis, $NM$
gPC	Generalized Polynomial Chaos
GPM	Gauss Pseudospectral Method
GPOPS	Gauss Pseudospectral Optimization Software
HBVP	Hamiltonian Boundary Value Problem
KKT	Karush-Kuhn-Tucker
kts	Knots
MCS	Monte-Carlo Simulation
N	Number of random input variables in gPC formulation
NLP	Nonlinear Programming
OC	Optimal Control
PDF	Probability Distribution Function
POK	Probability of Kill
RV	Random Variable
SDE	Stochastic Differential Equation
TO	Trajectory Optimization

### *Symbols*

$\alpha$	Collocation weight in gPC expansion
$\Gamma$	Finite domain of PDF
$\mu$	Mean or Expected Value of a probability distribution
$\Omega$	Space of all possible outcomes
$\phi_B$	Boundary conditions in optimal control problem formulation
$\Phi$	Basis polynomial element in the N-dimensional polynomial space
$\phi$	One-dimensional polynomial basis element
$\Phi_B$	Mayer cost functional term associated with boundary conditions
$\rho$	PDF of random variable
$\sigma$	Standard deviation of a probability distribution
$\tau$	Scaled time variable, $[-1, 1]$
$\theta$	Angle between $x_1$ axis and velocity vector, Rad

### *Superscripts*

$m$	Dimension of the control vector
$n$	Dimension of the state vector

## I. Introduction

**R**EAL-WORLD trajectory optimization (TO) and optimal control (OC) problems are typically solved using deterministic models, which are often simplified representations of a system's true physics that neglect several potential sources of uncertainty. Higher-fidelity models may be cost-prohibitive to develop, not feasible due to unknown physics, or difficult to use in engineering work. Additionally, virtually all sensors used to measure system states, for example speed, altitude, latitude, longitude, etc., are corrupted by noise thus inducing measurement errors that can degrade the validity of TO or OC solutions. The environment can also introduce randomness through phenomena such as wind gusts and turbulence. Therefore, all real-world problems have uncertainties that can make it difficult to find effective solutions and cause deterministic methods to fail to meet objectives.

Nonlinear optimization problems are typically too complex to use analytical solution techniques necessitating use of numerical methods. When stochastic elements are included, numerical methods become essentially the only way to solve these problems. This paper investigates combining two numerical techniques, the Gauss Pseudospectral Method (GPM) and Generalized Polynomial Chaos (gPC), to solve TO and OC problems with uncertain parameters. The GPM is a powerful deterministic numerical method for solving both linear and nonlinear optimal control problems and the gPC is an equally powerful method for solving stochastic differential equations. Effectively combining the two methods, drawing on the spectral accuracy of each, provides a new numerical approach for addressing nonlinear optimization problems where model, measurement, and environmental disturbance uncertainties may be included in the formulation and their effects assessed by constructing distribution functions showing dependence of the solutions on uncertain parameters.

This paper is organized as follows. Section II summarizes a general OC problem statement, Sections III and IV briefly describe the GPM and gPC, respectively, and Section V presents the hybrid algorithm combining the two methods. Section VI discusses application of the hybrid algorithm to a nonlinear TO problem, the results of which are presented in Section VII. The paper ends with conclusions and recommendations for future work in Section VIII.

## II. General Optimal Control Problem

**T**HE hybrid algorithm presented in this paper is a combination of numerical methods for solving TO and OC problems where uncertainties are present. Therefore, an appropriate starting point in the discussion is to briefly state the general problem, the traditional solution technique, and associated difficulties. The continuous-time Bolza-type problem can be generally stated as:<sup>1,2</sup> Determine the state,  $\mathbf{x}(\tau) \in \mathbb{R}^n$ , control,  $\mathbf{u}(\tau) \in \mathbb{R}^m$ , and the terminal time,  $t_f$  if unknown, to minimize the cost functional, ( $J$ ):

$$J = \Phi_B(\mathbf{x}(-1), \mathbf{x}(1)) + \frac{t_f - t_0}{2} \int_{-1}^1 g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) d\tau \quad (1)$$

subject to the differential (dynamic), boundary, and path constraints, respectively:

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \quad (2)$$

$$\phi_B(\mathbf{x}(-1), \mathbf{x}(1)) = \mathbf{0} \quad \in \mathbb{R}^q \quad (3)$$

$$\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{0} \quad \in \mathbb{R}^c \quad (4)$$

where:  $g$  defines the running cost in the Lagrange (integral) term in the cost functional,  $\mathbf{f}$  is the system dynamics,  $\Phi_B$  associates a cost penalty with the boundary conditions,  $\phi_B$  defines the boundary conditions, and  $\mathbf{C}$  places boundaries on acceptable state trajectories and control solutions. Note that Eqs. (1)-(4) have been written in the time variable,  $\tau$ , to be in the proper domain for GPM formulation, however transformations between the time domain,  $t \in [t_0, t_f]$ , and  $\tau \in [-1, 1]$ , are accomplished by:

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f + t_0}{2} \quad (5)$$

Classical OC theory (cf. Refs. 3,4, and 5) seeks to solve this problem indirectly by transforming it into a Hamiltonian Boundary Value Problem (HBVP) using Calculus of Variations theory and Pontryagin's Minimum Principle. The cost functional and constraints are adjoined via Lagrange multipliers to form the Hamiltonian function. The Hamiltonian is then used to derive first-order optimality conditions, known as the Euler-Lagrange equations, which results

in a set of coupled, generally nonlinear, differential equations that are solved simultaneously for the optimal state and costate trajectories and control inputs. Solving the Euler-Lagrange equations indirectly solves the original OC problem by finding the minimizing solution of the Hamiltonian, thereby minimizing the original cost functional.

Numerical solution techniques to solve the HBVP are usually necessary since analytical solutions are often impossible to find. This indirect approach yields highly accurate results while providing assurances that first-order optimality conditions are satisfied,<sup>6</sup> but explicit derivations of the Euler-Lagrange equations are required and prior knowledge of the activeness of inequality constraints (Eq. (4)) is necessary.<sup>7</sup> Numerical techniques applied to solve the HBVP generally require good initial guesses of the costates which are often difficult to make since they generally do not have direct physical interpretations.<sup>6,8</sup> These difficulties in solving the HBVP problem have motivated research in direct solution methods, such as the GPM, using pseudospectral discretization and nonlinear programming (NLP) techniques.

### III. Gauss Pseudospectral Method

THERE are several pseudospectral methods available to solve nonlinear optimal control problems such as the Legendre<sup>9,10</sup>, Chebyshev,<sup>11</sup> GPM, and Radau<sup>6,8,12</sup> pseudospectral methods. These methods are based on the same general procedure that parameterizes the states,  $\mathbf{x}(t)$ , and controls,  $\mathbf{u}(t)$ , using Lagrange interpolating polynomials, discretizes the constraints (Eqs. (2)-(4)) and the cost functional (Eq. (1)) using a quadrature rule, converts the differential dynamic constraints to algebraic constraints, and uses a NLP solver to find the solution to the resulting parameter optimization problem.<sup>1,6,8,10,11</sup> Several software packages, a sampling of which are listed in Ref. 6, have been developed to directly solve the continuous-time OC problem by discretizing and transcribing it into a NLP problem using this general procedure. Two common packages, both written for use in `Matlab`<sup>®</sup> and using SNOPT<sup>13</sup> as the NLP solver, are DIDO,<sup>14</sup> which implements the Legendre pseudospectral method developed in Ref. 10, and Gauss Pseudospectral Optimization Software (GPOPS),<sup>2</sup> which implements the GPM developed in Refs. 1, 6 and 8.

In this research, the GPM was chosen based on equivalence of the direct numerical solution and the indirect HBVP numerical solution as shown in Refs. 6 and 8. The key equations GPOPS uses to transform the continuous OC problem into a parameter optimization problem for SNOPT to solve are derived in Refs. 6 and 8, summarized in Refs. 1 and 2, and paraphrased below. Equations for the NLP optimality conditions, i.e. Karush-Kuhn-Tucker (KKT) conditions, and discussion of their equivalence to the HBVP optimality conditions can be found in Refs. 1, 2, 6 and 8.

The GPM process is to discretize the continuous equations (1)-(4) and transcribe them into a NLP problem. The process begins by choosing  $N$  Legendre-Gauss collocation points  $\in (-1, 1)$ , noting that the end points are not included in the set. The states,  $\mathbf{x}(\tau)$ , and controls,  $\mathbf{u}(\tau)$ , are approximated using  $N + 1$  Lagrange interpolating polynomials ( $L$ ) of the form:

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) L_i^x(\tau) \quad (6)$$

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^N \mathbf{U}(\tau_i) L_i^u(\tau) \quad (7)$$

where lower case symbols,  $\mathbf{x}(\tau)$  and  $\mathbf{u}(\tau)$ , denote the continuous state and control vectors while the capitalized symbols,  $\mathbf{X}(\tau)$  and  $\mathbf{U}(\tau)$ , represent the corresponding polynomial approximations. Additionally, superscripts  $x$  and  $u$  are used to distinguish between Lagrange polynomials representing the state and control approximations, respectively. The differential dynamic constraints (Eq. (2)) are approximated by differentiating Eq. (6):

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=0}^N \dot{x}(\tau_i) \dot{L}_i^x(\tau) \quad (8)$$

The derivatives of the Lagrange polynomials evaluated at the Legendre-Gauss collocation points can be written in a differentiation matrix,  $\mathbf{D}$ , which has dimension  $N \times N + 1$ , as given in Refs. 1 and 2. The dynamic constraints (Eq. (2)) are discretized and transformed to algebraic constraints suitable for the NLP solver using Eq. (6), (7), (8), and the differentiation matrix as follows:

$$\sum_{i=0}^N D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) = \mathbf{0} \quad (9)$$

where the counting indices are  $i = 0, \dots, N$  and  $k = 1, \dots, N$ .

For clarity, key shorthand notation used in this section is summarized by the following list:

- Discretized state vector at the  $k^{\text{th}}$  collocation point:  $\mathbf{X}_k \equiv \mathbf{X}(\tau_k) \in \mathbb{R}^n$  for  $k = 1, \dots, N$ .
- Discretized control vector at the  $k^{\text{th}}$  collocation point:  $\mathbf{U}_k \equiv \mathbf{U}(\tau_k) \in \mathbb{R}^m$  for  $k = 1, \dots, N$ .
- Discretized state vector at the initial time:  $\mathbf{X}_0 \equiv \mathbf{X}(-1)$ , where  $\tau = -1$  is not a collocation point.
- Discretized state vector at the final/terminal time:  $\mathbf{X}_f \equiv \mathbf{X}(1)$ , where  $\tau = 1$  is not a collocation point.

Since  $\mathbf{X}(\tau)$  and  $\mathbf{U}(\tau)$  are not collocated at the end points, it's necessary to approximate the terminal state,  $\mathbf{X}_f$ , using Gaussian quadrature:

$$\mathbf{X}_f = \mathbf{X}_0 + \frac{t_f - t_0}{2} \sum_{k=1}^N w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \quad (10)$$

The discretized form of the cost functional, obtained using Gaussian quadrature to approximate the integral term in Eq. (1), is written as:

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^N w_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \quad (11)$$

The last necessary items for the NLP are discretized forms of the boundary constraints (Eq. (3)) and path constraints (Eq. (4)), stated as:

$$\phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = \mathbf{0} \quad (12)$$

and:

$$\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \leq \mathbf{0} \quad (k = 1, \dots, N) \quad (13)$$

The NLP problem can now be stated as: find  $\mathbf{X}_k$  and  $\mathbf{U}_k$  that minimizes Eq. (11) subject to Eqs. (9), (10), (12), (13), and  $\mathbf{X}(\tau_0) = \mathbf{X}_0$ . GPOPS calls the SNOPT NLP solver to find the solution, which is an approximate solution to the continuous Bolza OC problem. In this paper, GPOPS is used as the deterministic solver in the gPC algorithm presented in the next two sections.

## IV. Generalized Polynomial Chaos

A survey of stochastic differential equation (SDE) solution methods, the most common being the brute-force *Monte-Carlo Simulation* (MCS) method, can be found in Ref. 15. MCS generates random samples of a stochastic variable taken from an assumed probability distribution function (PDF). Each random sample is subsequently inserted into the SDE, making it a deterministic problem that can be solved using deterministic differential equation solution methods. A set, or ensemble, of solutions is collected from which the statistical information, such as expected value (mean), variance, and covariance, is calculated. Convergence is checked at some arbitrary interval, after every 500 or 1000 samples for example, and is said to have been achieved when the change in expected value between successive intervals is below a desired tolerance. The solution to the problem is the combination of expected value, the most likely solution to the stochastic problem, and second moment variance and covariance information, which characterizes the distribution of the solution by showing how much the solution varies from the mean as a function of uncertain parameters and correlations between solution variables, respectively. MCS is relatively easy to implement using existing deterministic differential equation solvers, will eventually converge to a solution regardless of the PDF of the process, and is not limited by the amount of uncertainty in the variable. However, it's well known that the mean converges slowly, which implies high computational burden, especially in multi-dimensional problems. The gPC was formulated to preserve the ease of implementation by using sampled data and repetitive application of deterministic solvers, as in the MCS, while reducing computational burden by using collocation points as the sample points and a polynomial approximation to determine the solution.<sup>16</sup>

The gPC is a spectral approximation of a stochastic process used to solve stochastic differential equations, analogous to the deterministic spectral OC problem solution methods described in Section III. The beginnings of gPC theory can be traced back to 1938 when Norbert Wiener introduced "Homogeneous Chaos" theory, also known as Hermite Chaos. Wiener used Hermite (orthogonal) polynomials to approximate Gaussian processes.<sup>17</sup> It has been shown that the Hermite Chaos converges for any second-order random process with finite second moments (variances), and spectrally for Gaussian processes since the Hermite polynomial and Gaussian PDF weighting functions closely match.<sup>18</sup> Later work extended Wiener's Homogeneous Chaos by correlating orthogonal polynomial bases with other statistical distributions by matching weighting functions,<sup>18,19</sup> thus leading to generalized Polynomial Chaos theory for solving stochastic differential equations.

There are two forms of the gPC: the Galerkin and collocation forms. The Galerkin form is described in Refs. 16 and 18, detailed development of the collocation form can be found in Refs. 20,21,22 and 23, and rigorous convergence analysis is discussed in Refs. 20 and 24. Both types of gPC have been applied to solve SDEs in many engineering disciplines such as: analysis of nonlinear integrated circuit response,<sup>25</sup> fluids,<sup>26,27,28,29,30</sup> heat transfer,<sup>31,32,33</sup> and others referred to by the authors of these references.

The collocation form of the gPC was used in this work since it is more easily implemented. The Galerkin formulation can be cumbersome to implement since it results in a set of coupled partial differential equations in time, state space, and random space, which may be difficult or even impossible to derive, and generally requires coding problem-specific numerical solvers. In contrast, the collocation approach collapses multi-dimensional numerical integration into a single summation to solve a set of uncoupled differential equations that can usually be solved with existing numerical solvers.<sup>15</sup> Implementation of the collocation form of the gPC with GPOPS as the deterministic solver, generically referred to as gPC in the rest of this paper, will be described in Section V, but first it's necessary to summarize the key equations, paraphrased from Ref 16.

Consider a general system of differential algebraic equations in the following form:

$$\begin{cases} F(t, \mathbf{x}, \mathbf{x}', \dots, \mathbf{x}^{(l)}, \mathbf{p}) & = 0 \\ g(t_0, \mathbf{x}(t_0), \dots, \mathbf{x}^{(l)}(t_0), \mathbf{p}) & = 0 \end{cases} \quad t \in (t_0, T] \quad (14)$$

where  $F$  and  $g$  represent the SDE and initial conditions, respectively, and the key variables are:

- **State variables:**  $\mathbf{x} = (x_1, \dots, x_J) \in \mathbb{R}^J$ . Note that  $J = n$  in the GPM formulation.
- **Stochastic input parameters (random variables):**  $\mathbf{p} = (p_1, \dots, p_N) \in \mathbb{R}^N$
- **Outputs or observables:**  $\mathbf{z} = G(\mathbf{x}) = (z_1, \dots, z_K) \in \mathbb{R}^K$

The vector of random variables (RV),  $\mathbf{p}$ , is defined as an N-variate random vector whose elements are assumed to be independent. Further assume that the probability space is defined by the tuple  $(\Omega, \mathcal{A}, \mathcal{P})$ , where the space of all possible basic outcomes,  $\Omega$ , is equipped with  $\sigma$ -algebra,  $\mathcal{A}$ , which can be thought of as the space of possible events that can be derived from  $\Omega$ , and probability measure,  $\mathcal{P}$ .<sup>34</sup>

The key probabilistic variables in the gPC development are:

- **PDF:**  $\rho_i$  is the PDF of the random variable  $p_i(\omega)$ , for  $i = 1, \dots, N$  and  $\omega \in \Omega$ . The joint PDF is given by:

$$\rho(\mathbf{p}) = \prod_{i=1}^N \rho_i(p_i) \quad (15)$$

- **Finite domain of  $\rho_i$ :**  $\Gamma_i \equiv p_i(\Omega)$  transforms the infinite domain,  $\Omega$ , to the finite domain,  $\Gamma$ , which are intervals in  $\mathbb{R}$ , for  $i = 1, \dots, N$ . The total finite domain is defined as:

$$\Gamma \equiv \prod_{i=1}^N \Gamma_i \subset \mathbb{R}^N \quad (16)$$

One of the key elements of the gPC expansion is to approximate the random variables in  $\mathbf{p}$  using orthogonal polynomials. Each one-dimensional orthogonal polynomial space,  $W^{i,d_i}$ , is defined by:

$$W^{i,d_i} \equiv \{v : \Gamma_i \rightarrow \mathbb{R} : v \in \text{span}\{\phi_m(p_i)\}_{m=0}^{d_i}\} \quad (i = 1, \dots, N) \quad (17)$$

where the counting index,  $i$ , denotes the space associated with the random variable  $p_i$  and  $d_i$  specifies the order of the polynomial space. The one-dimensional polynomial basis set,  $\{\phi_m(p_i)\}$ , has  $d_i + 1$  elements and satisfies the orthogonality conditions:

$$\int_{\Gamma_i} \rho_i(p_i) \phi_m(p_i) \phi_n(p_i) dp_i = \left[ \int_{\Gamma_i} \rho_i \phi_m^2 dp_i \right] \delta_{mn} = h_m^2 \delta_{m,n} \quad (18)$$

where scaling of the polynomial basis elements is chosen such that  $h_m^2 = 1$  for all  $m$ .<sup>15</sup> The N-variate orthogonal polynomial space of polynomials of total degree of at most  $P$  is constructed by taking the tensor product of the N one-dimensional orthogonal polynomial bases, written as:

$$W_N^P \equiv \bigotimes_{|\mathbf{d}| \leq P} W^{i,d_i} \quad (19)$$



where the relationship between  $d_i$  and  $P$  is given by:

$$|\mathbf{d}| = \sum_{i=1}^N d_i \leq P \quad (20)$$

and satisfies the orthogonality conditions:

$$\int_{\Gamma} \Phi_m(\mathbf{p}) \Phi_n(\mathbf{p}) \rho(\mathbf{p}) d\mathbf{p} \equiv \mathbb{E}[\Phi_m(\mathbf{p}) \Phi_n(\mathbf{p})] = \delta_{mn} \quad (21)$$

Note that the lower case  $\phi$  is used to indicate the one-dimensional polynomial basis element and the upper case  $\Phi$  is used to represent the N-dimensional polynomial basis element resulting from the tensor product. The effect of Eq. (20) is to select a subset of basis elements from the tensor product space defined by Eq. (19) in order to reduce computational burden, however the entire set of basis elements can be used,<sup>15</sup> in which case  $M$  is equal to the total number of tensor product basis elements.

The approximation of the solution,  $z$ , as a function of the random parameters,  $\mathbf{p}$ , can be written as:

$$\mathbb{P}_N^P z \equiv z_N^P(\mathbf{p}) = \sum_{m=1}^M \hat{z}_m \Phi_m(\mathbf{p}) \quad (22)$$

where:

$$M = \binom{N+P}{N} \quad (23)$$

or is the total number of tensor product elements if the entire basis set is used in the approximation. The projection operator in Eq. (22),  $\mathbb{P}_N^P$ , projects the discrete probability space  $\Gamma$  onto the polynomial space  $W_N^P$ . The expansion coefficients,  $\hat{z}_m$ , are determined by:

$$\hat{z}_m = \mathbb{E}[z(\mathbf{p}) \Phi_m(\mathbf{p})] = \int_{\Gamma} z(\mathbf{p}) \Phi_m(\mathbf{p}) \rho(\mathbf{p}) d\mathbf{p} \quad (m = 1, \dots, M) \quad (24)$$

Another key element in the gPC method is to approximate the integral in Eq. (24) using Gaussian quadrature. To apply quadrature, a set of collocation points and quadrature weights must be chosen. For each random dimension,  $\Gamma_i$ ,  $i = 1, \dots, N$ ,  $q_i$  collocation points and weights are chosen based on the quadrature rule. For example, if Hermite orthogonal polynomial bases are chosen to approximate a Gaussian distribution (see Refs. 15 and 35), then Gauss-Hermite quadrature is an appropriate quadrature rule and will be most accurate using Gauss-Hermite nodes and weights. The N-dimensional grid with  $Q$  collocation nodes and weights is obtained by taking the tensor product of the one-dimensional node and weight sets. Alternatively, a sparse grid can be generated to reduce the total number of tensor product nodes while providing accurate integral approximation using the Smolyak algorithm, first published in Ref. 36 and included, along with other sparse grid schemes, in Ref. 37. In general, the tensor product grid works well for low-dimensional problems, but suffers the *curse of dimensionality* as  $N$  gets larger ( $N > 5$ ) making sparse grids a more attractive choice for reducing computations.<sup>15,20</sup>

The collocation approximation of Eq. (24) based on the quadrature rule chosen is given by:

$$\hat{z}_m = \sum_{j=1}^Q z(\mathbf{p}_j) \Phi_m(\mathbf{p}_j) \alpha_j; \quad (m = 1, \dots, M) \quad (25)$$

where  $\mathbf{p}_j$  denotes the  $j^{\text{th}}$  collocation node of the random vector  $\mathbf{p} = (p_{1,j}, \dots, p_{N,j})$  and  $\alpha_j$  denotes the associated quadrature weight, thus defining the set of collocation nodes and weights,  $\{\mathbf{p}_j, \alpha_j\}_{j=1}^Q$ . Note that  $z(\mathbf{p}_j)$  is the deterministic solution using the  $j^{\text{th}}$  sample of the random vector.

The solution to the SDE given by Eq. (22) is a distribution function that approximates the outputs (observables) as functions of the random inputs and can be evaluated for any given sample point or set of points  $\mathbf{p} \in \Gamma$  to see individual solutions and how they vary with changes in  $\mathbf{p}$ . The expansion coefficients, Eq. (25), can be used to calculate the distribution's statistical properties; expected value, variance, and covariance, to find the most likely solution to the stochastic problem, characterize how the solution varies with uncertain parameters, and describe the dependencies between solution variables. The equations for expected value, variance, and covariance are derived in Ref. 15 and are stated below:

$$\mathbb{E}(\mathbf{z}(t)) \approx \mathbb{E}(\mathbf{z}_N^P(t)) = \int \left[ \sum_{m=1}^M \hat{\mathbf{z}}_m(t) \Phi_m(\mathbf{p}) \right] \rho(\mathbf{p}) d\mathbf{p} = \hat{\mathbf{z}}_1(t) \quad (26)$$



$$\text{var}[\mathbf{z}(t)] \approx \sum_{m=2}^M [\hat{\mathbf{z}}_m^2(t)] \quad (27)$$

$$\text{cov}[z_i(t), z_j(t)] \approx \sum_{m=2}^M [\hat{z}_{i,m}(t)\hat{z}_{j,m}(t)] \quad (28)$$

The material presented in this section provides the basis for a general solution procedure where the GPM and gPC can be combined to address stochastic optimization problems, which is presented in the next section.

## V. Hybrid Algorithm

**T**HE algorithm presented in this section and shown in figure 1 is an adaptation of the algorithm published in Ref. 16 that was originally written to solve SDEs using initial value and boundary value problem differential equation routines to solve the set of deterministic sample problems used in stochastic computations.

Xiu's algorithm has been adapted in this work to be applicable to TO and OC problems by using GPOPS in place of the initial value and boundary value problem solvers because the Gauss pseudospectral method implemented in GPOPS assures spectrally accurate deterministic solutions, if they exist, that satisfy optimality conditions and minimize the performance index while meeting specified constraints. Therefore, GPOPS, or similar pseudospectral method solvers, provides the best possible approximations to the  $Q$  deterministic problems, while avoiding the difficulties of formulating and solving the HBVP, to be used in the gPC approximation.

**Step #1:** Choose a set of collocation nodes and quadrature weights  $\{\mathbf{p}_j, \alpha_j\}_{j=1}^Q$  in the finite probability space  $\Gamma$ . The collocation nodes should be chosen to be consistent with both the quadrature rule evaluating the expansion coefficients in Eq. (24) and the polynomial basis elements  $\Phi_m$ . There are many **Matlab**<sup>®</sup> functions available for download on the MathWorks<sup>™</sup> website to generate quadrature nodes and weights and descriptions of various quadrature rules can be found in Refs. 38, 39 and 40.

**Step #2:** Using each of the  $Q$  sample points,  $\mathbf{p}_j$ , solve the deterministic OC problem using GPOPS, or any other pseudospectral method software, and build the observables array  $\mathbf{z} = [\mathbf{x}(t), \mathbf{u}(t), t_f, J(\mathbf{x}, \mathbf{u})]^T$ . Other variables of interest, such as costates and the Hamiltonian, can be included in the observables array.

**Step #3:** Evaluate the gPC expansion coefficients using Eq. (25).

**Step #4:** Build the gPC approximation using Eq. (22). This output of the gPC algorithm is a function of the random inputs  $\mathbf{p}$ .

**Step #5:** Evaluate the statistics of the gPC approximate solution using Eqs. (26), (27), and (28).

The algorithm combining the GPM with the gPC was applied to a challenging nonlinear TO problem with random inputs affecting the cost functional described in the next section.

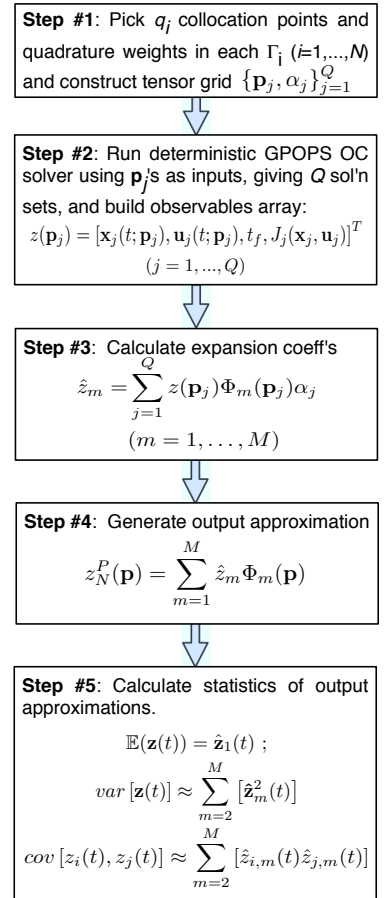


Figure 1. Hybrid algorithm combining GPM and gPC methods.

## VI. Trajectory Optimization Problem

A trajectory optimization problem, notionally sketched in figure 2, is considered to demonstrate the ability of the hybrid algorithm to quantify the effects of uncertain parameters on an optimal trajectory solution.

The scenario was designed to represent an aircraft travelling to a desired target location through an environment where there are potential risks of lethal engagements. The objective was to find the optimal path in a two-dimensional space that takes a vehicle from an initial position to a target location while minimizing the probability that it will be *killed* by the threats whose locations are uncertain. It's assumed that at some point in time the center locations were perfectly known but have possibly moved in the time between intelligence gathering and mission planning according to an assumed probability distribution. The deterministic TO problem will be described first, which will form the basis for the subsequently discussed stochastic problem.

### VI.A. Deterministic Problem Formulation

To begin adding details to the notional scenario in figure 2, a two-dimensional grid of 50 x 50 nautical miles (NM) was chosen. Three threats with effective ring diameters of approximately 20NM were arbitrarily placed in the grid at (10NM, 30NM), (25NM, 15NM), and (35NM, 30NM). The target was chosen to be located at (25NM, 50NM). Threat and target locations can be easily changed to accommodate user-supplied scenario specifications. Likewise, the diameters of the threat rings can be adjusted and are not constrained to be uniformly sized. Units of nautical miles and knots (kts) were used since they are familiar to aircrews and mission planners who have interest in this type of problem.

The vehicle dynamics are generically represented with the *Dubins* model in Eq. (29). This model was used to avoid adding the complexity of developing and coding a model for a specific vehicle's dynamics, which is not necessary to demonstrate the algorithm's performance.

$$\begin{aligned}\dot{x}_1(t) &= V \cos(\theta(t)) \\ \dot{x}_2(t) &= V \sin(\theta(t)) \\ \dot{\theta}(t) &= u(t)\end{aligned}\tag{29}$$

The state and control vectors are defined as  $\mathbf{x}(t) = [x_1(t), x_2(t), \theta(t)]^T \in \mathbb{R}^3$  and  $\mathbf{u}(t) = [\dot{\theta}(t)] \in \mathbb{R}^1$ , respectively. The vehicle speed ( $V$ ) is assumed to be a constant 470 knots (kts), chosen based on a mission profile of a generic large aircraft, and the control is bounded as:

$$|u(t)| \leq \frac{V}{R_{min}}\tag{30}$$

where the minimum turn radius ( $R_{min}$ ) based on the constant speed is estimated to be 11,326 feet. The initial and terminal boundary conditions are:

$$\begin{aligned}\mathbf{x}(t_0) &= \mathbf{x}(-1) = [0, 0, 1.1071]^T \\ \mathbf{x}(t_f) &= \mathbf{x}(1) = [25, 50, free]^T\end{aligned}\tag{31}$$

where  $t_f$  is unspecified. Lastly, admissible paths are restricted to lie in the boundaries of the grid, stated in the form of Eq. (4) as:

$$[-x_1(\tau), (x_1(\tau) - 50), -x_2(\tau), (x_2(\tau) - 50)]^T \leq \mathbf{0}\tag{32}$$

The most important element in setting up the TO problem is to appropriately choose a cost functional in the form of Eq. (1) that leads to the trajectory that allows the aircraft to reach the target while minimizing the vehicle's exposure to lethal threats and not prohibiting it from passing through higher threat areas of the space. The integrand of the Lagrange term in Eq. (1) must be chosen to define the areas of the space shown in figure 2 that pose the greatest risk to the vehicle. In an obstacle avoidance problem, the threat shapes would be treated as path constraints, as investigated by Gong et al.<sup>41</sup> and Lewis et al.,<sup>42</sup> making any trajectory that passes through those areas of the space inadmissible.

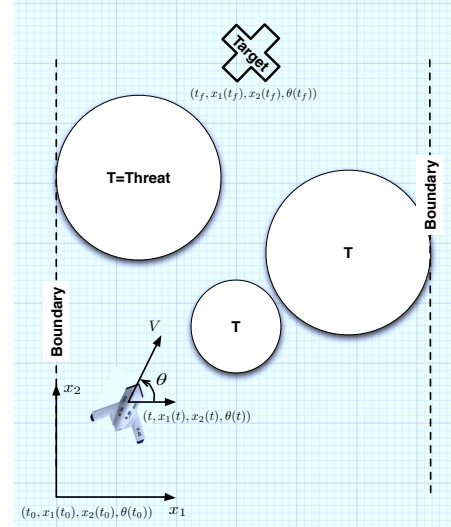


Figure 2. Notional sketch of the trajectory optimization problem

Numerical solvers would only return solutions that totally avoid those areas of the space or output a message stating that no optimal solution exists that satisfies these constraints. However, in this case, the threats are treated as regions of the space that represent varying probabilities that the vehicle will be killed, not as hard obstacles that should be totally avoided. The threat rings represent likelihood, or probabilities, that travelling into those areas would result in the vehicle being eliminated. Random elements, discussed later in this section, are introduced through uncertainty of the exact center locations of the threat rings, making it possible for the threat rings to overlap and impossible to completely avoid the threats. Therefore, the integrand of the cost functional,  $g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f)$ , should be something that describes the probabilities of kill (POK) in the space, assuming that the highest probabilities will be near the center of the threat rings and decrease with distance away from the center. Minimizing such a functional will allow the vehicle to pass through regions of the space occupied by the threat rings while finding the least threatening path to the target. A bi-variate Gaussian PDF for each threat was used to describe the high-threat regions of the two dimensional space, written as:

$$g_i(x_1(\tau), x_2(\tau), \tau; t_0, t_f) = \frac{1}{2\pi\sigma_{i,x1}\sigma_{i,x2}} \exp\left(-\frac{1}{2}\left[\frac{(x_1(\tau) - \mu_{i,x1})^2}{\sigma_{i,x1}^2} + \frac{(x_2(\tau) - \mu_{i,x2})^2}{\sigma_{i,x2}^2}\right]\right) \quad (33)$$

where the mean values,  $\mu_{i,x1}$  and  $\mu_{i,x2}$ , and the standard deviations,  $\sigma_{i,x1}$  and  $\sigma_{i,x2}$ , define the center and variation in lethality of the  $i^{th}$  threat in the  $x_1$  and  $x_2$  directions for  $i = 1, \dots, 3$ . Figure 3 depicts the high POK regions using the previously stated center locations and  $\sigma_{i,x1} = \sigma_{i,x2} = 5NM$  to create rings with  $20NM$  effective diameters.

One final consideration is necessary in defining the cost functional. Looking at figure 3, it is obvious that multiple trajectories may exist that have equal minimum POK, meaning that a unique solution may not exist, resulting in GPOPS failure to return a solution. To minimize this possibility, the Mayer term in the cost functional should be chosen to drive the software to return the minimum POK path with shortest travel time to the target. Including the Mayer term in the cost functional gives:

$$J(x_1(\tau), x_2(\tau), \tau; t_0, t_f) = (10^{-5})t_f + \frac{t_f - t_0}{2} \int_{-1}^1 \sum_{i=1}^3 g_i(x_1(\tau), x_2(\tau), \tau; t_0, t_f) d\tau \quad (34)$$

where the weighting factor on  $t_f$  is used to make sure the Mayer and Lagrange terms are on the same order of magnitude to avoid having the minimum time requirement dominate the solution.

Finally, the TO problem can be stated as: Minimize  $J$  (Eq. (34)) subject to the dynamics (Eq. (29)), control constraint (Eq. (30)), boundary conditions (Eq. (31)), and path constraints (Eq. (32)). This is a continuous-time Bolza-type TO problem that is nonlinear in both the cost functional and the state dynamics, with fixed final state and free final time, and constrained boundary conditions and control input. The solution to the problem described thus far will be referred to as the *deterministic* solution, i.e the noise-free solution, which was found using GPOPS and is shown in figure 3. The mean solution of the stochastic variant discussed in the next sub-section will be compared to this solution and should closely match it since the uncertainties will be assumed to be zero mean.

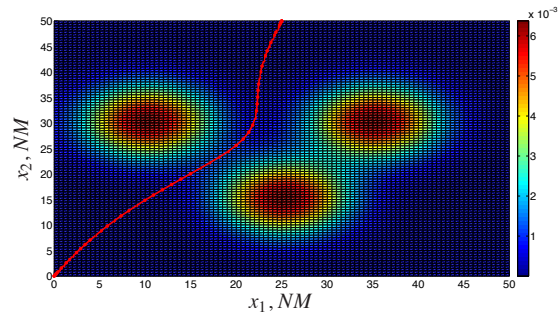


Figure 3. Threat ring locations and deterministic solution to the trajectory optimization problem

## VI.B. Stochastic Problem Formulation

The stochastic problem is formulated in this section by modifying the deterministic version in the previous sub-section following the development in Section IV. Uncertainty is added to the deterministic problem by introducing three random input parameters,  $A_1$ ,  $A_2$ , and  $A_3$ , and making the following substitutions indicated by the ( $\rightarrow$ ) into the Lagrange term of the cost functional (Eqs. (33) and (34)).

$$\mu_{i,x1} \rightarrow \mu_{i,x1} + A_i \quad (35)$$

$$\mu_{i,x2} \rightarrow \mu_{i,x2} + A_i \quad (36)$$

The integrand of the modified cost functional becomes:

$$g_i(x_1(\tau), x_2(\tau), \tau; t_0, t_f) = \frac{1}{2\pi\sigma_{i,x1}\sigma_{i,x2}} \exp\left(-\frac{1}{2}\left[\frac{(x_1(\tau) - (\mu_{i,x1} + A_i))^2}{\sigma_{i,x1}^2} + \frac{(x_2(\tau) - (\mu_{i,x2} + A_i))^2}{\sigma_{i,x2}^2}\right]\right) \quad (37)$$

resulting in the modified cost functional:

$$J(x_1(\tau), x_2(\tau), \tau; A_i, t_0, t_f) = (10^{-5})t_f + \frac{t_f - t_0}{2} \int_{-1}^1 \sum_{i=1}^3 g_i(x_1(\tau), x_2(\tau), \tau; A_i, t_0, t_f) d\tau \quad (38)$$

where the  $A_i$ 's are assumed to be independent Gaussian RVs with zero mean,  $\mu = 0$ , and standard deviation,  $\sigma$ , of 1. The random inputs thus defined and written into Eqs. (37) and (38) introduce uncertainties on center locations of the threats, transforming  $J$ ,  $\mathbf{x}$ ,  $u$ , and  $t_f$  into RVs whose solutions will have statistical properties.

It should be noted that the objective of a stochastic OC problem is typically to find the control signal and state trajectories that minimize the expected value of the cost functional, written as:

$$J = \mathbb{E} \left[ (10^{-5})t_f + \frac{t_f - t_0}{2} \int_{-1}^1 \sum_{i=1}^3 g_i(x_1(\tau), x_2(\tau), \tau; A_i, t_0, t_f) d\tau \right] \quad (39)$$

However, in this paper, quantification of the effects of uncertain parameters on  $J(\mathbf{x}, u)$ ,  $\mathbf{x}(t)$ ,  $u(t)$ , and  $t_f$  solutions is sought since there is no way to control the uncertainties or minimize their effects. So, the cost functional as expressed by Eqs. (37) and (38) is used when solving each of the  $Q$  sampled deterministic problems, and the set of solutions is used to construct gPC approximations of the states, control, cost, and final time as functions of the random inputs to assess the effects of those uncertainties.

The dynamics (Eq. (29)), control bounds (Eq. (30)), boundary conditions (Eq. (31)), and path constraints (Eq. (32)) remain the same in the stochastic problem.

Following the development in Section IV, the following definitions are made to identify the key details needed to apply the gPC method:

- **State variables:**  $\mathbf{x} = (x_1, x_2, \theta) \in \mathbb{R}^3$  ( $J = n = 3$ )
- **Stochastic input parameters (random variables):**  $\mathbf{p} = (A_1, A_2, A_3) \in \mathbb{R}^3$  ( $N = 3$ ).
- **Outputs or observables:**  $\mathbf{z} = (x_1, x_2, \theta, u, t_f, J) \in \mathbb{R}^6$  ( $K = 6$ )

The PDFs (Gaussian) of  $A_1$ ,  $A_2$ , and  $A_3$  are denoted by  $\rho_1(A_1)$ ,  $\rho_2(A_2)$ , and  $\rho_3(A_3)$ , respectively. The joint PDF is given by Eq. (15) as:

$$\rho(\mathbf{p}) = \prod_{i=1}^3 \rho_i(p_i) = \rho_1(A_1)\rho_2(A_2)\rho_3(A_3) \quad (40)$$

While continuous Gaussian distributions are defined on the sample space  $\Omega = (-\infty, \infty) \times (-\infty, \infty) \times (-\infty, \infty)$ , the finite sample space is defined by the collocation points. Since Gaussian PDFs were used in this case, Hermite polynomial basis functions collocated at Gauss-Hermite quadrature points were chosen for the gPC expansion as suggested in Refs. 15 and 35. It's important to note that the statistical version of the Hermite polynomials was used, where the weighting function is  $\exp[-\frac{(x-\mu)^2}{2\sigma^2}]$ , instead of the typical  $\exp[-x^2]$ , which means that the traditional Gauss-Hermite quadrature points need to be scaled by a factor of  $\sqrt{2\sigma^2}$  and shifted by adding  $\mu$  and the quadrature weights scaled by  $(\sqrt{\pi})^{-1}$ . Recall there are  $q$  points in each random dimension, which are calculated by finding the roots of the  $q^{th}$  Hermite polynomial,  $H_q$ , and scaled accordingly. The number of collocation points in the three random dimensions,  $q_1$ ,  $q_2$ , and  $q_3$ , was chosen to be 7, resulting in  $Q = 343$  collocation nodes and the finite probability space  $[-4, 4] \times [-4, 4] \times [-4, 4]$  as shown in figure 4.

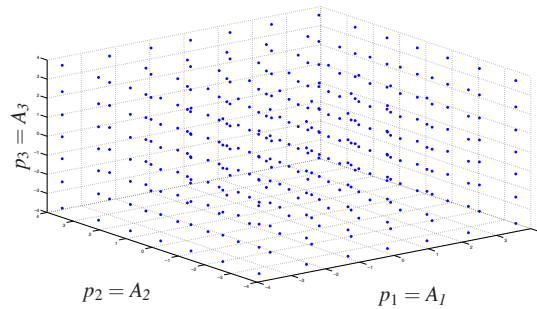


Figure 4. Collocation points: Tensor grid of 343 points (7 in each dimension) used as random inputs  $A_1, A_2$ , and  $A_3$ .

The one-dimensional polynomial spaces were chosen, as given in Eq. (17), to be defined by the following polynomial basis sets:

$$\begin{aligned}\{\phi_1\}_{d_1=0}^5 &= \{H_0, \dots, H_5\} \\ \{\phi_2\}_{d_2=0}^5 &= \{H_0, \dots, H_5\} \\ \{\phi_3\}_{d_3=0}^5 &= \{H_0, \dots, H_5\}\end{aligned}\tag{41}$$

where  $H_i$  denotes the Hermite polynomial of the  $i^{\text{th}}$  order. Thus, the 3-dimensional polynomial space, defined using the tensor product in Eq. 19, is:

$$\{\phi_1\}_{d_1=0}^5 \otimes \{\phi_2\}_{d_2=0}^5 \otimes \{\phi_3\}_{d_3=0}^5\tag{42}$$

Selecting the entire set of basis functions yields 216 total basis elements, where each one-dimensional basis set is at most fifth order and the highest order of the tensor product set is 15. The number of basis elements was chosen, all of which are used in computations, to provide a balance between computational burden and accuracy of the solution when compared to MCS.

This section has presented all of the details necessary to implement the hybrid algorithm steps described in Section V. The results of applying the algorithm to the sample problem considered in this paper are presented in the next section.

## VII. Results

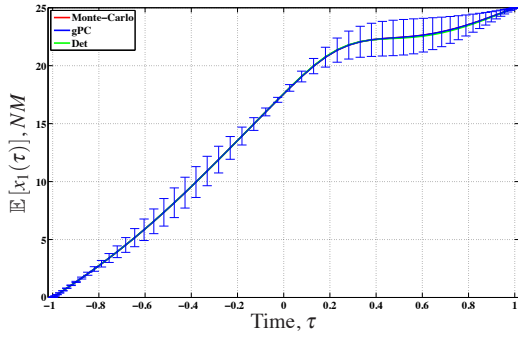
RESULTS of applying the hybrid GPM-gPC algorithm to the trajectory optimization problem are presented in this section. The expected value solutions and variances are plotted for  $x_1(\tau)$ ,  $x_2(\tau)$ ,  $x_3(\tau)$ , and  $u(\tau)$  and tabulated for  $J$  and  $t_f$ . Covariances between the states  $x_1(\tau)$  and  $x_2(\tau)$ ,  $x_1(\tau)$  and  $x_3(\tau)$ , and  $x_2(\tau)$  and  $x_3(\tau)$ , as well as between the states and control,  $x_1(\tau)$  and  $u(\tau)$ ,  $x_2(\tau)$  and  $u(\tau)$ , and  $x_3(\tau)$  and  $u(\tau)$ , are also shown. The results for  $u(\tau)$  are not necessary since this is a TO problem, but are included to show agreement between MCS and gPC approximations and that the gPC solution satisfies the bounded control constraints given by Eq. (30). Since this is a free final time problem, expected values, variances, and covariances are plotted versus the time vector  $\tau$ . Stochastic computations evaluating the gPC output function (Eq. (22)), gPC expansion coefficients (Eq. (25)), and statistical quantities derived from the expansion coefficients are not possible in the time domain since the time points in the  $Q$  sampled solutions are not the same once GPOPS makes the transformation from the  $\tau$  domain to the time domain using Eq. (5). The  $\tau$  vectors, however, remain consistent throughout regardless of the differing terminal times and allows for stochastic computations using a consistent time reference. In a fixed final time problem, these stochastic computations can be performed using the time domain.

Figure 5 on the next page shows the mean solutions for  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ , and  $u(t)$ . There are three curves on each of the sub-plots. The first is an MCS estimation of the mean solution, which converged after 25K iterations using random inputs ( $A_1, A_2, A_3$ ) drawn from Gaussian distributions with  $\mu_i = 0$  and  $\sigma_i = 1$ , requiring more than eight hours of processing on a MacBook Pro computer with a 2.2 gigahertz quad-core processor and eight gigabytes of memory. The ensemble of 25K optimal GPOPS trajectories was used to calculate the expected values (mean) shown on in the figure. The second curve in the sub-figures is the solution to the deterministic OC problem, given by Eqs. (29)-(34) or equivalently by the stochastic formulation using Eqs. (29)-(32), (37), and (38) with  $A_1 = A_2 = A_3 = 0$ , which serves as another check of the mean solution. Since the stochastic inputs are unbiased (zero mean), it was expected that the gPC mean solution would be close to the deterministic one, which is confirmed by figure 5 on the following page. This check was possible by virtue of the sample problem formulation and may not be valid in more general problems. The last curve is the hybrid GPM-gPC algorithm approximation of the expected value, recalling that it is given by the first gPC expansion coefficient as stated in Eq. (26). It's interesting to note that the hybrid GPM-gPC method using GPOPS required just over seven minutes to solve the 343 TO problems and use the data to construct the gPC approximation of the solution. Bars indicating two standard deviations are included to quantify how much the solution varies from the mean. These plots show that the gPC and MCS closely match each other, which both match the deterministic solution as expected, and the desired terminal conditions  $x_1(t_f)$  and  $x_2(t_f)$  are satisfied.

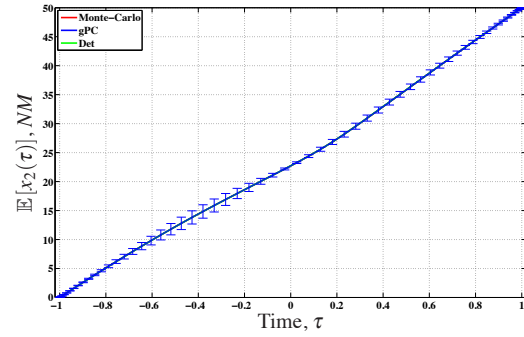
Figure 6 on page 14 shows the variances of the hybrid algorithm solutions of  $x_1(\tau)$ ,  $x_2(\tau)$ ,  $x_3(\tau)$ , and  $u(\tau)$  using Eq. (27). These plots show close correlation between they hybrid algorithm and MCS variance estimates and that desired terminal conditions,  $x_1(t_f)$  and  $x_2(t_f)$ , are satisfied since the variances are zero at the terminal time with the hybrid method performing the computations more than 90% faster than the MCS.

Covariance estimations further characterize the distributions of the stochastic solution by providing indications of dependence between the output variables. Figure 7 on page 15 shows the covariance estimates provided by the hybrid

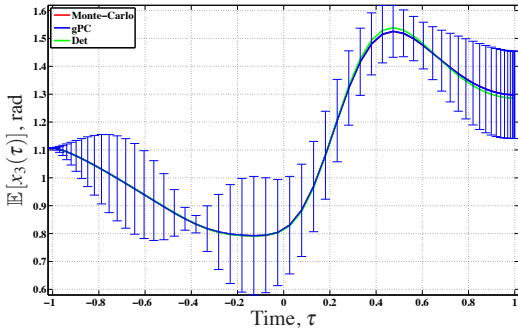




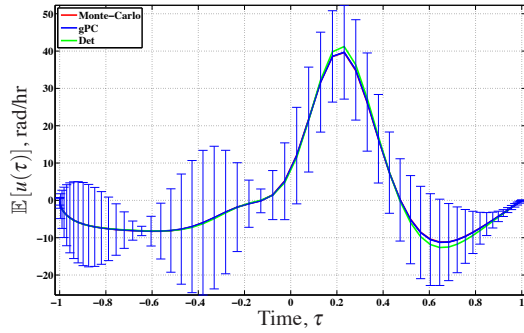
(a) Expected Value of  $x_1(\tau)$



(b) Expected Value of  $x_2(\tau)$



(c) Expected Value of  $x_3(\tau)$



(d) Expected Value of  $u(\tau)$

Figure 5. MCS, deterministic, and gPC expected value estimates of states,  $x_1(\tau)$ ,  $x_2(\tau)$ , and  $x_3(\tau)$ , and control,  $u(\tau)$ .

algorithm and compares them with MCS estimates. As with the mean and variance estimates, the hybrid algorithm results closely match the MCS results, indicating that the algorithm is generating reasonable solutions. The covariance plots also show that the state variables and control outputs are dependent even though the random inputs are assumed to be independent, which is as expected since the variables are related through the equations of motion (Eq. (29)).

Expected value and variance results for estimating the cost,  $J$ , and final time,  $t_f$ , are given in table 1. The hybrid

Table 1. MCS and gPC Expected Value and Variance of  $J$  and  $t_f$

	Expected Value Algorithm <sup>c</sup>	Expected Value MCS <sup>d</sup>	Difference	Variance Algorithm	Variance MCS)	Difference
$J^a$	$1.6129\text{E}^{-5}$	$1.6131\text{E}^{-5}$	$2.0\text{E}^{-9}$	$1.4998\text{E}^{-12}$	$1.5359\text{E}^{-12}$	$3.61^{-14}$
$t_f^b$	0.1232	0.1232	0.0	$7.3970\text{E}^{-7}$	$8.4716\text{E}^{-7}$	$1.0746\text{E}^{-7}$

<sup>a</sup> Deterministic solution of  $J$ :  $1.6164\text{E}^{-5}$

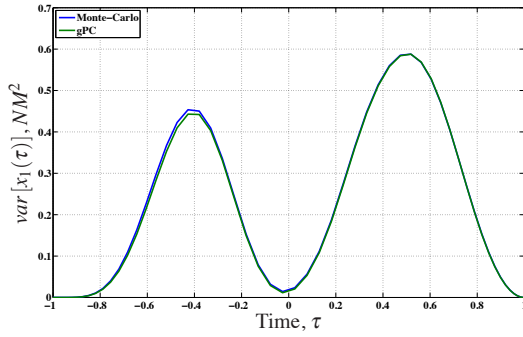
<sup>b</sup> Deterministic solution of  $t_f$ : 0.1230 hours

<sup>c</sup> Computation time: approximately 7 minutes

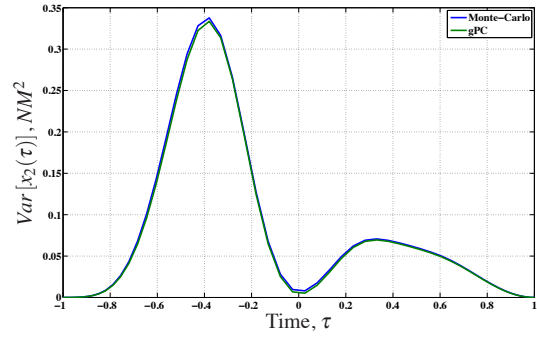
<sup>d</sup> Computation time: 8+ hours

algorithm and MCS approximations for the expected value and variance closely match each other as shown by the small differences given in the table and are actually equal, to four decimal places, in expected value of the final time. Furthermore, the expected values of both  $J$  and  $t_f$  agree with the deterministic solution.

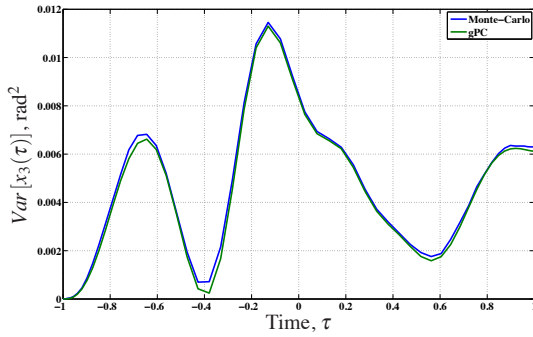
The expected value plots (figure 5), along with the variance plots (figure 6 on the following page) and covariance plots (figure 7 on page 15) give the most likely solution to the stochastic trajectory optimization problem and describe the nature of the output distributions by quantifying the affects of the Gaussian random parameters on the output so-



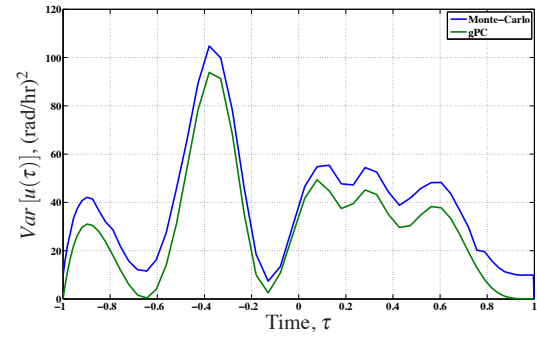
(a) Variance of  $x_1(\tau)$



(b) Variance of  $x_2(\tau)$



(c) Variance of  $x_3(\tau)$



(d) Variance of  $u(\tau)$

Figure 6. MCS and gPC variance estimates of states,  $x_1(\tau)$ ,  $x_2(\tau)$ , and  $x_3(\tau)$ , and control,  $u(\tau)$ .

lutions. The agreement of the hybrid algorithm and MCS results shown in this section provides confidence that the gPC-based hybrid algorithm provides accurate approximations of the output distributions while reducing computational workload.

## VIII. Conclusions and Future Work

A hybrid algorithm combining GPM and gPC methods was presented and applied to a nonlinear stochastic TO problem with uncertain parameters included in the cost functional. The hybrid method extends current well-established methods to solve SDEs to be applicable to solve TO and OC problems by combining a powerful and accurate pseudospectral method to solve the sampled deterministic TO and OC problems with a pseudospectral method that characterizes the statistical effects of the uncertain parameters on the solution. Using GPOPS as the deterministic solver provided accurate deterministic solutions at the sample points, thus is well suited to be used in place of standard differential equation solvers. The results demonstrated that the method is able to generate solutions to the stochastic TO and OC problems by determining the expected value solutions for the state variables, control, cost, terminal time and characterize the probabilistic information about the solutions that agrees with MCS results, thus characterizing how the optimal solution changes with uncertainty.

Further investigation will focus on improving numerical efficiency of the hybrid algorithm by seeking a “rule-of-thumb” for selecting appropriate numbers of polynomial basis elements that balance accuracy with computational burden. Sparse grids will also be incorporated to further improve efficiency by reducing the number of sample solutions necessary to quantify the effects of uncertain parameters on the solution, which will be critical in higher-dimensional problems. Calling GPOPS 343 times to solve the sampled OC problems was not overly time consuming, but keeping in mind that GPOPS uses a NLP solver, which is an iterative routine in itself, to calculate state and control solutions that meet optimality conditions, it is not difficult to surmise that as the number of states, controls, and random inputs grow



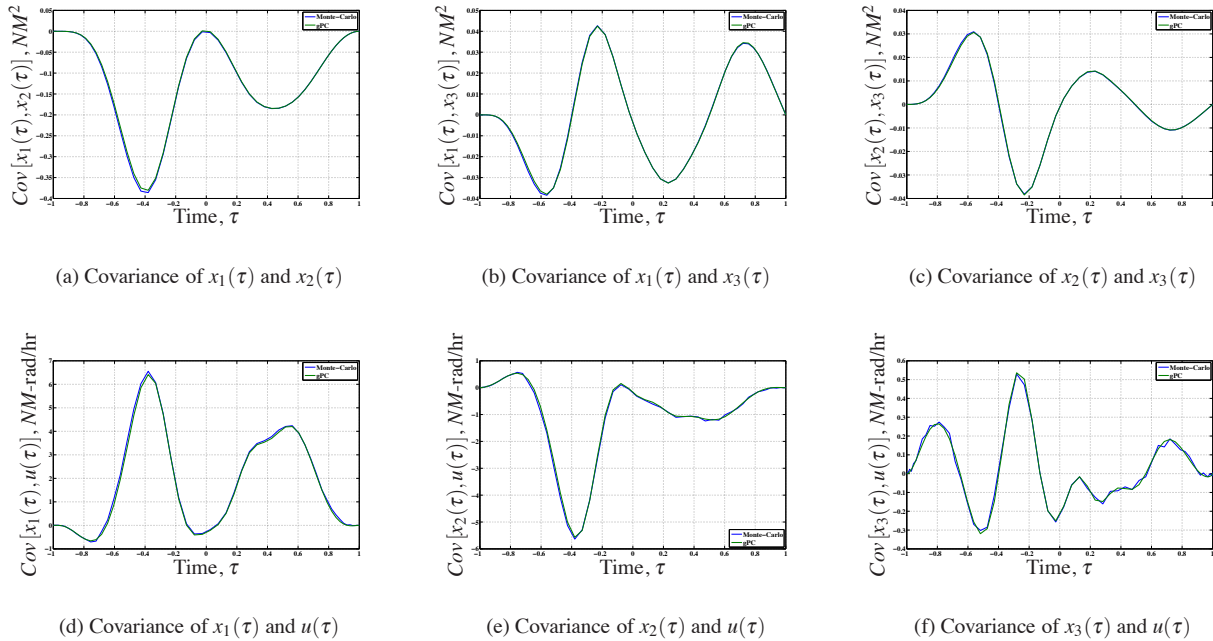


Figure 7. MCS and gPC covariance estimates between the states,  $x_1(\tau)$ ,  $x_2(\tau)$ , and  $x_3(\tau)$ , and between the states and control,  $u(\tau)$ .

the computational burden will increase exponentially. Additionally, problems will be considered where the random elements are not assumed to have the same PDFs, leading to a mix of polynomial basis sets in the gPC approximation that will provide insight into how the interactions between random input distributions affect the distributions of the observables estimates. Lastly, future research will seek to further develop the algorithm to address problems with time-varying random parameters, such as wind gusts and sensor noise, that could eventually lead to hybrid algorithm application to real- or near-real-time state estimation and control problems.

This paper was a first-look at combining the gPC algorithm with the GPM to solve trajectory optimization problems with stochastic inputs. The results show great potential, but also highlight the need for much further work.

## References

- <sup>1</sup>Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440.
- <sup>2</sup>Rao, A. V., Benson, D., Huntington, G. T., Franconin, C., Darby, C. L., Patterson, M., and Sanders, I., "User's Manual for GPOPS Version 3.0: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Pseudospectral Methods," 2010.
- <sup>3</sup>Kirk, D. E., *Optimal Control Theory: An Introduction*, Dover Publications, Mineola, NY, April 2004.
- <sup>4</sup>Bryson, A. E., *Dynamic Optimization*, Addison-Wesley, Menlo Park, CA, June 1999.
- <sup>5</sup>Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The Mathematical Theory of Optimal Processes*, Wiley, New York, 1962.
- <sup>6</sup>Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.
- <sup>7</sup>von Stryk, O. and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, , No. 37, 1992, pp. 357–373.
- <sup>8</sup>Benson, D., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.
- <sup>9</sup>Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The pseudospectral Legendre method for discretizing optimal control problems," *Automatic Control, IEEE Transactions on*, Vol. 40, No. 10, 1995, pp. 1793–1796.
- <sup>10</sup>Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277.
- <sup>11</sup>Fahroo, F. and Ross, I. M., "Direct Trajectory Optimization by Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166.
- <sup>12</sup>Garg, D., Patterson, M. A., Franconin, C., Darby, C. L., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems Using a Radau Pseudospectral Method," *Computational Optimization and Applications*, 2009.
- <sup>13</sup>Gill, P. E., Murray, W., and Saunders, M. A., "User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming," 2008.

- <sup>14</sup>“DIDO, A Pseudospectral Method for Solving Optimal Control Problems,” 2006, MIT Online Course Lecture (16.323 Lecture 17).
- <sup>15</sup>Xiu, D., “Fast Numerical Methods for Stochastic Computations: A Review,” *Communications in Computational Physics*, Vol. 5, No. 2-4, 2009, pp. 242–272.
- <sup>16</sup>Xiu, D., “Efficient Collocational Approach for Parametric Uncertainty Analysis,” *Communications in Computational Physics*, Vol. 2, No. 2, 2007, pp. 293–309.
- <sup>17</sup>Wiener, N., “The Homogeneous Chaos,” *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.
- <sup>18</sup>Xiu, D. and Karniadakis, G. E., “The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations,” *SIAM Journal on Scientific Computing*, Vol. 24, No. 2, 2002, pp. 619–644.
- <sup>19</sup>Askey, R. and Wilson, J. A., “Some Basic Hypergeometric Orthogonal Polynomials That Generalize Jacobi Polynomials,” *Memoirs of the American Mathematical Society*, Vol. 54, No. 319, 1985, pp. 1–55.
- <sup>20</sup>Babuska, I., Nobile, F., and Tempone, R., “A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data,” *SIAM Journal on Numerical Analysis*, Vol. 45, No. 3, 2007, pp. 1005–1034.
- <sup>21</sup>Eldred, M. S., “Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design,” *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, May 4-7, 2009 2009.
- <sup>22</sup>Xiu, D. and Hesthaven, J. S., “High-Order Collocation Methods for Differential Equations with Random Inputs,” *SIAM Journal on Scientific Computing*, Vol. 27, No. 3, 01/07 2006, pp. 1118–1139.
- <sup>23</sup>Pulch, R., “Polynomial Chaos for Boundary Value Problems of Dynamical Systems,” *Applied Numerical Mathematics (to appear)*.
- <sup>24</sup>Nobile, F., Tempone, R., and Webster, C., “The analysis of a sparse grid stochastic collocation method for partial differential equations with high-dimensional random input data,” Tech. Rep. SAND2007-8093, Dec 01 2007.
- <sup>25</sup>Tao, J., Zeng, X., Cai, W., Su, Y., Zhou, D., and Chiang, C., “Stochastic Sparse-grid Collocation Algorithm (SSCA) for Periodic Steady-State Analysis of Nonlinear System with Process Variations,” *ASP-DAC '07: Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 474–479.
- <sup>26</sup>Gottlieb, D. and Xiu, D., “Galerkin Method for Wave Equations with Uncertain Coefficients,” *Communications in Computational Physics*, Vol. 3, No. 2, 2008, pp. 505–518.
- <sup>27</sup>Matre, O. P. L., Knio, O. M., Najm, H. N., and Ghanem, R. G., “A Stochastic Projection Method for Fluid Flow: I. Basic Formulation,” *Journal of Computational Physics*, Vol. 173, No. 2, 11/1 2001, pp. 481–511.
- <sup>28</sup>Matre, O. P. L., Reagan, M. T., Najm, H. N., Ghanem, R. G., and Knio, O. M., “A Stochastic Projection Method for Fluid Flow: Random Process,” *Journal of Computational Physics*, Vol. 181, No. 1, 9/1 2002, pp. 9–44.
- <sup>29</sup>Mathelin, L. and Hussaini, M. Y., “A Stochastic Collocation Algorithm for Uncertainty Analysis,” Tech. Rep. NASA/CR-2003-212153, 2003.
- <sup>30</sup>Xiu, D. and Karniadakis, G. E., “Modeling uncertainty in flow simulations via generalized polynomial chaos,” *Journal of Computational Physics*, Vol. 187, No. 1, 5/1 2003, pp. 137–167.
- <sup>31</sup>Wan, X. and Karniadakis, G. E., “Multi-Element Generalized Polynomial Chaos for Arbitrary Probability Measures,” *SIAM Journal on Scientific Computing*, Vol. 28, No. 3, 11 2006, pp. 901–928.
- <sup>32</sup>Xiu, D. and Karniadakis, G. E., “A new stochastic approach to transient heat conduction modeling with uncertainty,” *International Journal of Heat and Mass Transfer*, Vol. 46, No. 24, 11 2003, pp. 4681–4693.
- <sup>33</sup>Ghanem, R., “Stochastic Finite Elements With Multiple Random Non-Gaussian Properties,” *Journal of Engineering Mechanics*, Vol. 125, No. 1, 1999, pp. 26–40.
- <sup>34</sup>Ghanem, R., “Ingredients for a general purpose stochastic finite elements implementation,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 168, No. 1-4, 1/6 1999, pp. 19–34.
- <sup>35</sup>Xiu, D., *Numerical Methods for Stochastic Computations*, Princeton University Press, Princeton, NJ, 2010.
- <sup>36</sup>Smolyak, S. A., “Quadrature and Interpolation Formulas for Tensor Products of Certain Classes,” *Soviet Mathematics - Doklady*, Vol. 4, 1963, pp. 240–243.
- <sup>37</sup>Gerstner, T. and Griebel, M., “Numerical Integration Using Sparse Grids,” *Numerical Algorithms*, Vol. 18, No. 3-4, 1998, pp. 209–234.
- <sup>38</sup>Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, New York, 1996.
- <sup>39</sup>Boyd, J. P., *Chebyshev and Fourier Spectral Methods*, Dover, New York, 2001.
- <sup>40</sup>Trefethen, L. N., *Spectral Methods in MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, 2001.
- <sup>41</sup>Gong, Q., Kang, W., Bedrossian, N., Fahroo, F., Sekhvat, P., and Bollino, K., “Pseudospectral Optimal Control for Military and Industrial Applications,” *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 4128–4142.
- <sup>42</sup>Lewis, L., Ross, I., and Gong, Q., “Pseudospectral motion planning techniques for autonomous obstacle avoidance,” *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 5997–6002.